

CA-ADS[®]

DSECT Reference
15.0



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

Second Edition, October 2001

© 2001 Computer Associates International, Inc.
All rights reserved.

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

| | |
|-------------------------------|-----|
| How to Use this Manual | vii |
| 1.0 #AACTDS | 1 |
| 1.1 #ADBDS | 2 |
| 1.2 #ADEDS | 4 |
| 1.3 #ADSDBST | 5 |
| 1.4 #AFGDS | 6 |
| 1.5 #AGBDS | 7 |
| 1.6 #AGLDS | 8 |
| 1.7 #AGN1CDS | 9 |
| 1.8 #AGNDS | 10 |
| 1.9 #AGRDS | 11 |
| 1.10 #AGWADS | 12 |
| 1.11 #AIRDS | 18 |
| 1.12 #AMHDS | 20 |
| 1.13 #AMRDS | 21 |
| 1.14 #ANCBDS | 22 |
| 1.15 #APGDS | 23 |
| 1.16 #APLGDS | 25 |
| 1.17 #APREDS | 28 |
| 1.18 #ARDDS | 29 |
| 1.19 #ARSDS | 30 |
| 1.20 #ASBDS | 31 |
| 1.21 #ASCDS | 33 |
| 1.22 #ASPDS | 34 |
| 1.23 #ASQDS | 37 |
| 1.24 #ASRDS | 38 |
| 1.25 #ATCDS | 39 |
| 1.26 #AWERRS | 40 |
| 1.27 #BDSDS | 41 |
| 1.28 #BLEDS | 42 |
| 1.29 #BLPDS | 49 |
| 1.30 #BMRBHDS | 50 |
| 1.31 #BPADS | 51 |
| 1.32 #BPXDS | 55 |
| 1.33 #BRPLDS | 58 |
| 1.34 #BSBDS | 62 |
| 1.35 #BSPDS | 64 |
| 1.36 #CMEDS | 66 |
| 1.37 #CMEEQU | 80 |
| 1.38 #DGBDS | 82 |
| 1.39 #DIBDS | 84 |
| 1.40 #DNBDS | 86 |
| 1.41 #DRNDS | 87 |
| 1.42 #DSDWADS | 88 |
| 1.43 #DWADS | 89 |
| 1.44 #DXBDS | 90 |
| 1.45 #EFHDRDS | 92 |

| | |
|---------------|-----|
| 1.46 #EFMASDS | 93 |
| 1.47 #EFXDEDS | 94 |
| 1.48 #EXCDS | 95 |
| 1.49 #EXCFNDS | 96 |
| 1.50 #EXEDS | 97 |
| 1.51 #FDBDS | 98 |
| 1.52 #FDEDS | 100 |
| 1.53 #FHEDS | 102 |
| 1.54 #FMDDDS | 104 |
| 1.55 #FMIOWRK | 105 |
| 1.56 #FSRTABL | 106 |
| 1.57 #GABDS | 107 |
| 1.58 #GARDS | 109 |
| 1.59 #GCMDS | 111 |
| 1.60 #GFDDDS | 112 |
| 1.61 #GFEDS | 116 |
| 1.62 #GFRDS | 117 |
| 1.63 #GLEDS | 118 |
| 1.64 #GMDDDS | 119 |
| 1.65 #GMEDS | 120 |
| 1.66 #GPRDS | 121 |
| 1.67 #GRDDDS | 123 |
| 1.68 #GREDS | 126 |
| 1.69 #GRLDS | 129 |
| 1.70 #GSADS | 130 |
| 1.71 #GSSDDS | 142 |
| 1.72 #GTCDS | 143 |
| 1.73 #GVCDS | 144 |
| 1.74 #GWADS | 145 |
| 1.75 #HCHDS | 150 |
| 1.76 #HVTDS | 152 |
| 1.77 #INCDS | 153 |
| 1.78 #LGSDS | 154 |
| 1.79 #LMTDS | 156 |
| 1.80 #LNTDS | 157 |
| 1.81 #LOBDS | 158 |
| 1.82 #MAPTDS | 159 |
| 1.83 #MASKDS | 160 |
| 1.84 #MDTDS | 161 |
| 1.85 #MSQDS | 162 |
| 1.86 #MTEDS | 163 |
| 1.87 #MXDEDS | 164 |
| 1.88 #MXSHDS | 165 |
| 1.89 #NAMEDS | 166 |
| 1.90 #OCBDS | 167 |
| 1.91 #OCCLDS | 169 |
| 1.92 #OFDDDS | 170 |
| 1.93 #OFTDS | 171 |
| 1.94 #OTBXDS | 172 |
| 1.95 #PEXCDS | 174 |
| 1.96 #PFLDS | 175 |

| | |
|----------------|-----|
| 1.97 #PICDS | 177 |
| 1.98 #PLFDS | 178 |
| 1.99 #PMEDS | 179 |
| 1.100 #PRBDS | 180 |
| 1.101 #PSADS | 181 |
| 1.102 #PWADS | 194 |
| 1.103 #RACTDS | 195 |
| 1.104 #RBBDS | 196 |
| 1.105 #RDBDS | 198 |
| 1.106 #RDEDS | 199 |
| 1.107 #REXDS | 201 |
| 1.108 #RKEDS | 203 |
| 1.109 #RLODS | 204 |
| 1.110 #RPEDS | 205 |
| 1.111 #RSEDS | 206 |
| 1.112 #RSPDS | 207 |
| 1.113 #RSRTABL | 208 |
| 1.114 #RSTDs | 209 |
| 1.115 #RVBDS | 210 |
| 1.116 #SACTDS | 211 |
| 1.117 #SDEDS | 212 |
| 1.118 #SELDS | 213 |
| 1.119 #SFEDS | 214 |
| 1.120 #SFTDS | 215 |
| 1.121 #SREDS | 216 |
| 1.122 #SRTDS | 219 |
| 1.123 #SSEDS | 220 |
| 1.124 #STHDS | 221 |
| 1.125 #SWRDS | 226 |
| 1.126 #SYADS | 228 |
| 1.127 #TAEDS | 230 |
| 1.128 #TATUDS | 231 |
| 1.129 #TFEDS | 232 |
| 1.130 #URNDS | 233 |
| 1.131 #VDBDS | 234 |
| 1.132 #WKBCTL | 237 |
| Index | X-1 |

How to Use this Manual

What this manual is about

This manual is intended to serve as comprehensive reference for all major DSECTs occurring in the CA-ADS environment in 15.0.

How this manual is organized

DSECTs appear in alphabetical order of name. Each DSECT is listed in the table of contents that follows this preface.

Each DSECT is also indexed by both a description of its purpose and by name.

Related documentation

- *CA-IDMS Dictionary DSECT Reference Guide*
- *CA-IDMS DSECT Reference Guide*
- *CA-IDMS Security Administration*
- *CA-IDMS DML Reference - Assembler*

1.0 #AACTDS

```

COPY #AACTDS
*****
***          AACT: AREA ACTIVITY CONTROL BLOCK
***          ***
***          AACT IS A DSECT THAT DESCRIBES THE ACTIVITY IN A CERTAIN ***
***          SUBSCHEMA AREA. IT IS BUILT IN ADSOGNP1, ADSOBGPS,
***          AND ADSOBGPL BEFORE THE CALL TO ADSOGEXC. IT IS ANCHORED ***
***          OFF OF THE GSS BY GSSAACTA.
***          ***
***          THE VALUES IN THE FIRST HALFWORD OF EACH ENTRY REPRESENT ***
***          THE DECIMAL VALUES OF EACH TYPE OF AREA COMMAND.
***          ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------------|--------------------------------------|
| 000000 | AACT DSECT | 00:02:18 11/09/84 |
| 000000 | AFRICNT DS F'00030000' | ALL FORMS OF FIND WITHIN AREA |
| 000004 | AACTELEN EQU *-AACT | LENGTH OF ONE ENTRY IN AACT |
| 000008 | AUPDCNT DS F'00360000' | UPDATE VALUE & COUNT |
| 00000C | ARETCNT DS F'00370000' | RETRIEVAL VALUE & COUNT |
| 000010 | APRUPCNT DS F'00380000' | PROTECTED UPDATE VALUE & COUNT |
| 000014 | APRRECNT DS F'00390000' | PROTECTED RETRIEVAL & COUNT |
| 000018 | AEXRECNTR DS F'00400000' | EXCLUSIVE RETRIEVAL & COUNT |
| 00001C | AEXUPCNT DS F'00410000' | EXCLUSIVE UPDATE & COUNT |
| 000020 | AOBTPCNT DS F'00430000' | OBTAIN WITHIN AREA & COUNT |
| 000024 | AOBKPCNT DS F'00630000' | OBT KEEP IN AREA & COUNT *PHH84310* |
| 000028 | AFNKPCTN DS F'00230000' | FND KEEP IN AREA & COUNT *PHH84310* |
| 00002C | AKEEPCNT DS F'00060000' | KEEP OR KEEP LONG IN AREA *PHH84313* |
| 000030 | AACCPCTN DS F'00150000' | ACCEPT WITHIN AREA & COUNT*PHH84310* |
| 0000C | AACTLEN EQU AACTLEN/AACTELEN | LENGTH OF AACT |
| | | # OF ENTRIES=LEN OF TBL/ENTRY LEN |

1.1 #ADBDs

```
COPY #ADBDs
*****
***      ADB IS THE DSECT THAT DEFINES THE APPLICATION DEFINITION      ***
***      BLOCK WHICH REPRESENTS AN ADS APPLICATION.                      ***
***      ***
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|--|
| 000000 | ADB DSECT | 12/05/89 15:05:13 09/04/90 |
| 000000 | ADBDID DS CL4 | "ADB*" |
| 000004 | ADBNNAME DS CL8 | APPLICATION NAME |
| 00000C | ADBDATET DS CL8 | APPLICATION GEN DATE |
| 000014 | ABDTIME DS CL8 | APPLICATION GEN TIME |
| 00001C | ADBDPRTD DS CL8 | DEFAULT PRINT DESTINATION |
| 000024 | ADBTSKA DS F | OFFSET TO TASK/TOP FUNCTION TABLE |
| 000028 | ADBRDEA DS F | OFFSET TO RDE TABLE FOR GLOBAL RECORDS |
| 00002C | ADBARERA DS F | OFFSET TO APRE TABLE FOR RESPONSES |
| 000030 | ADBRKEA DS F | OFFSET TO RESPONSE KEY TABLE |
| 000034 | ADBFDEA DS F | OFFSET TO FIRST FDE (FUNCTION TABLE) |
| 000038 | ADBMTEA DS F | OFFSET TO MENU EXTENSIONS TABLE |
| 00003C | ADBLITA DS F | OFFSET TO LITERAL POOL |
| 000040 | ADBBTMA DS F | OFFSET TO BITMAP POOL |
| 000044 | ADBAVER DS H | APPLICATION VERSION NUMBER |
| 000046 | ADBSFCIX DS H | SIGNON FUNCTION INDEX |
| 000048 | ADBNFNC DS H | NUMBER OF FUNCTIONS (FDE'S) |
| 00004A | ADBRBML DS H | LENGTH OF APRE TABLE BITMAPS |
| 00004C | ADBKBMIL DS H | LENGTH OF RKE TABLE BITMAPS |
| 00004E | ADBNREC DS H | NUMBER OF GLOBAL RECORDS (RDE'S) |
| 000050 | ADBNRPR DS H | NUMBER OF PASSED RECORDS |
| 000052 | ADBMAXRS DS H | MAXIMUM NUMBER OF RESPONSES |
| 000054 | ADBNNTKS DS H | NUMBER OF INITIATING TASK CODES |
| 000056 | ADBECL DS X | SECURITY CLASS |
| 000057 | ADBPRL DS X | DEFAULT PRINTER CLASS |
| 000058 | ADBDfmt DS X | DEFAULT DATE FORMAT CODE |
| | PRINT NOGEN | |
| | ADBSORQ #FLAG X'80' | ON = SIGNON REQUIRED |
| | ADBSOOP #FLAG X'40' | ON = SIGNON IS OPTIONAL |
| | ADBMALL #FLAG X'20' | ON = MENUS CONTAIN ALL RESPONSES |
| | ADBMSEC #FLAG X'10' | ON = MENUS ARE SECURITY TAILORED |
| | ADBFast #FLAG X'08' | ON = FAST MODE IS DEFAULT |
| 000059 | ADBFAG DS X | ADB FLAG BYTE |
| | ADBroAU #FLAG X'40' | ON = RESTART OPTION AUTOMATIC |
| | ADBroPR #FLAG X'20' | ON = RESTART OPTION PROMPT |
| | ADBroMA #FLAG X'10' | ON = RESTART OPTION MANUAL |
| | *NOTE: IF ADBroAU, ADBroPR AND ADBroMA ARE ALL OFF THEN THE | |
| | * RESTART OPTION IS NO. | |
| | ADBonLN #FLAG X'08' | ON = APPLICATION WILL RUN ONLINE |
| | ADBBTCH #FLAG X'04' | ON = APPLICATION WILL RUN BATCH |
| | *NOTE: IF ADBonLN AND ADBBTCH ARE BOTH SET, THE APPLICATION | |
| | * WILL RUN IN ANY MODE. | |
| 00005A | ADBFAG2 DS X | ADB FLAG BYTE 2 |
| | PRINT GEN | |
| | ADBADSC #FLAG X'80' | ON = APPLICATION ADSC LEVEL OR ABOVE |
| 00005B | ADBADSCI DS 0XL1 | |
| | 000080 ADBADSCM EQU X'80' | |
| | ADBEVA #FLAG X'40' | ON = APPLICATION STORED IN DEV DICT |
| 00005B | ADBEVAI DS 0XL1 | |
| | 000040 ADBDEVAM EQU X'40' | |
| 00005B | ADBFAG3 DS X | ADB FLAG BYTE 3 |
| 00005C | ADBPRIA DS F | OFFSET TO RDE INDEX LIST, PASSED RECS |

| | | | | |
|--------|---------|-----|-------|-------------------------|
| 000060 | ADBDESC | DS | CL32 | APPLICATION DESCRIPTION |
| 000080 | | DS | 7F | RESERVED |
| 0009C | ADBLEN | EQU | *-ADB | LENGTH OF ADB HEADER |

1.2 #ADEDS

```

COPY #ADEDS
*****
*** ADE IS A DSECT WHICH DEFINES AN ARGUMENT DESCRIPTION ELEMENT ***
*** WHICH IS FOUND IN A DML COMMAND CME. A DML COMMAND CME MAY ***
*** CONTAIN FROM ZERO TO THREE ADE'S. EACH ADE DESCRIBES AN ***
*** ARGUMENT TO BE PASSED TO IDMS TO EXECUTE THE COMMAND. EACH ***
*** ADE IS ASSOCIATED WITH ONE OF THE FIRST THREE CME FLAGS. ***
*** THE FLAG DEFINES THE FORMAT AND CONTENT TYPE OF THE ADE. ***
*****
```

| <u>Offset</u> | <u>Value</u> |
|--|---|
| 000000 | ADE DSECT * ** FORMAT ONE ADE (ONE FULLWORD) * |
| 000000 | ADEFMT1 DS F RDE INDEX, SSAN OFFSET OR NUMERIC |
| 000004 | ADEF1LEN EQU *-ADE LENGTH OF FORMAT ONE ADE * ** FORMAT TWO ADE (TWO HALFWORDS) * |
| 000004 | ORG ADEFMT1 |
| 000000 | ADELITO DS H LITPOOL OFFSET |
| 000002 | ADELITLN DS H LENGTH OF LITERAL |
| 000004 | ADEF2LEN EQU *-ADE LENGTH OF FORMAT TWO ADE * ** FORMAT THREE ADE (FOUR HALFWORDS) * |
| 000004 | ORG ADEFMT1 |
| 000000 | ADEVRETO DS H VRE TABLE OFFSET |
| 000002 | ADEDTAOF DS H DATA OFFSET |
| 000004 | ADEDTYPE DS H DATA TYPE (SEE XDE DATA TYPE EQUATES) |
| 000006 | ADEDLEN DS H DATA LENGTH |
| 000008 | ADEF3LEN EQU *-ADE LENGTH OF FORMAT THREE ADE |
| ***** | |
| * A FORMAT THREE ADE MAY BE FOLLOWED BY AN XDE STACK TO BE USED TO * | |
| * EXTRACT A FIELD FROM A RECORD, TO MOVE A FIELD FROM THE IRA TO * | |
| * A USER FIELD OR TO SIMPLY POINT AT A USER FIELD IN A RECORD. THE * | |
| * USE OF THE XDE STACK DIFFERS DEPENDING ON THE REQUIREMENTS OF * | |
| * THE CME IN WHICH IT IS FOUND. * | |
| ***** | |

1.3 #ADSDBST

```
COPY #ADSDBST
*****
***      #ADSDBST: ADS DATABASE STATISTICS BLOCK
***      ***
***      ADSDBST DSECT IS NEW WITH R10.2. IT HOLDS
***      @ACCEPT'ED DATABASE STATISTICS FOR CCTL, RUN2, AND RUNB
***      ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|------------------|----------------------------|
| 000000 | ADSDBST | DSECT | |
| 000000 | ADSDBCBC | DS OF | |
| 000000 | | DS CL76 | DB STATISTICS |
| 00004C | ADSDBRI | DS F | DB RUN UNIT ID |
| 000050 | | DS CL20 | DB STATISTICS |
| 00064 | ADSDBLEN | EQU *-ADSDBCBC | LENGTH OF STATISTICS BLOCK |

1.4 #AFGDS

```

COPY #AFGDS
*****
***      AFG:  ADS APPLICATION GENERATOR GENERAL FUNCTION RECORD ***
***      AFG IS A DSECT THAT DESCRIBES THE FIELDS IN THE          ***
***      ADSO-FUNC-GENERAL RECORD USED BY THE ADS APPLICATION    ***
***      GENERATOR.                                              ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---|-------------------------|---|
| 000000 | AFG DSECT | 11/27/89 16:25:36 09/04/90 |
| 000000 | AFGSEQ DS F | MODCMT-038 SEQUENCE NUMBER |
| 000004 | AFGID DS CL8 | FUNCTION NAME |
| 00000C | AFGDESC DS CL32 | DESCRIPTION |
| 00002C | AFGINVEE DS CL8 | DIALOG/PGM INVOKED BY FCN |
| 000034 | AFGXITD DS CL8 | USER EXIT DIALOG NAME |
| 00003C | AFGRSPPG DS H | # RESPONSES/PAGE FOR MENU FCN |
| 00003E | AFGDRESP DS CL8 | DEFAULT RESPONSE |
| 000046 | AFGINVTP DS X | VOKEE TYPE |
| 000047 | AFGFCNTP DS X | FUNCTION TYPE |
| 0000C4 | AFGDIALG EQU C'D' | DIALOG |
| 0000D7 | AFGPGM EQU C'P' | PROGRAM |
| 0000D4 | AFGMENU EQU C'M' | MENU |
| 0000E2 | AFGSYSTM EQU C'S' | SYSTEM |
| 0000C9 | AFGINTRN EQU C'I' | INTERNAL |
| 0000E4 | AFGUNDEF EQU C'U' | UNDEFINED |
| * | | |
| * USER DEFINED MENUS ARE SPECIFIED BY: | | |
| * AFGFCNTP = AFGMENU | | |
| * AFGINVTP = AFGDIALG | | |
| * | | |
| 000048 | DS X | UNUSED |
| 000049 | AFGMENSG DS X | MENU SIGNON ('Y' OR 'N') |
| 00004A | AFGNHDGL DS X | NUMBER OF HEADING LINES (PIC 9) |
| 00004B | AFGFCNDL DS X | FUNCTION DESCRIPTOR LENGTH ('S' OR 'L') |
| 00004C | DS 68X | FILLER |
| 000090 | AFGTTYPE DS F | TYPE OF MODCMT-038 RECORD |
| 000094 | AFGLEN EQU *-AFG | LENGTH OF DSECT |
| 00025 | AFGLENF EQU (*-AFG)/4 | LENGTH OF DSECT IN FULLWORDS |

1.5 #AGBDS

```
COPY #AGBDS
*****
***      AGB:  ADS APPLICATION GENERATOR BITMAP ELEMENT
***      ***
***  AGB IS A DSECT FOR THE INTERNAL CONTROL BLOCK USED TO
***  HOLD A BITMAP WHICH IS DESTINED FOR INCLUSION IN THE
***  BITMAP POOL WHEN IT IS CONSTRUCTED FOR THE ADB.
***      ***
*****
```

Offset Value

| | | | |
|--------|----------|-------------------|------------------------------|
| 000000 | AGB | DSECT | 11:53:46 04/04/85 |
| 000000 | AGBNXTA | DS A | ADDR OF NEXT AGB |
| 000004 | AGBPREVA | DS A | ADDR OF PREVIOUS AGB |
| 000008 | AGBSIZE | DS F | SIZE OF BITMAP |
| 00000C | AGBBPOFF | DS F | ACTUAL OFFSET IN BITMAP POOL |
| 00010 | AGBLLEN | EQU *-AGB | LENGTH OF AGB |
| 00004 | AGBLLENF | EQU ((*-AGB+3)/4) | LENGTH OF AGB IN WORDS |
| 000010 | AGBBTM | DS 0X | START OF BITMAP |

1.6 #AGLDS

```
COPY #AGLDS
*****
***      AGL:  ADS APPLICATION GENERATOR LITERAL ELEMENT
***      AGL IS A DSECT FOR THE INTERNAL CONTROL BLOCK USED TO
***      HOLD A LITERAL WHICH IS DESTINED FOR INCLUSION IN THE
***      LITERAL POOL WHEN IT IS CONSTRUCTED FOR THE ADB.
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------------|-------------------------------|
| 000000 | AGL DSECT | 11:54:22 04/04/85 |
| 000000 | AGLNXTA DS A | ADDR OF NEXT AGL |
| 000004 | AGLPREVA DS A | ADDR OF PREVIOUS AGL |
| 000008 | AGLSIZE DS F | SIZE OF LITERAL |
| 00000C | AGLLPOFF DS F | ACTUAL OFFSET IN LITERAL POOL |
| 00010 | AGLLEN EQU *-AGL | LENGTH OF AGL |
| 00004 | AGLLENF EQU ((*-AGL+3)/4) | LENGTH OF AGL IN WORDS |
| 000010 | AGLLIT DS 0X | START OF LITERAL |

1.7 #AGN1CDS

```

COPY #AGN1CDS
*****
*          FUNCTION, OPERATION, AND OCCURRENCE CODES FOR      *
*          CALLING ADSOAGN1                                     *
*****
Offset  Value
00006C    0001AGNFNC1 DC  H'1'           AGN1 CALL - FUNCTION 1
00006E    0002AGNFNC2 DC  H'2'           AGN1 CALL - FUNCTION 2
000070    0003AGNFNC3 DC  H'3'           AGN1 CALL - FUNCTION 3
000072    0004AGNFNC4 DC  H'4'           AGN1 CALL - FUNCTION 4
000074    0005AGNFNC5 DC  H'5'           AGN1 CALL - FUNCTION 5
000076    0006AGNFNC6 DC  H'6'           AGN1 CALL - FUNCTION 6
000078    0007AGNFNC7 DC  H'7'           AGN1 CALL - FUNCTION 7
00007A    0008AGNFNC8 DC  H'8'           AGN1 CALL - FUNCTION 8
00007C    0009AGNFNC9 DC  H'9'           AGN1 CALL - FUNCTION 9
00007E    000AAGNFNC10 DC  H'10'          AGN1 CALL - FUNCTION 10
000080   000BAGNFNC11 DC  H'11'          AGN1 CALL - FUNCTION 11
000082   000CAGNFNC12 DC  H'12'          AGN1 CALL - FUNCTION 12
000084   000DAGNFNC13 DC  H'13'          AGN1 CALL - FUNCTION 13
000086   000EAGNFNC14 DC  H'14'          AGN1 CALL - FUNCTION 14
000088   000FAGNFNC15 DC  H'15'          AGN1 CALL - FUNCTION 15
000C1    AGNOPADD EQU  C'A'           AGN1 CALL - ADD OPERATION
000D4    AGNOPMOD EQU  C'M'           AGN1 CALL - MODIFY OPERATION
000C4    AGNOPDEL EQU  C'D'           AGN1 CALL - DELETE OPERATION
000D9    AGNOPRET EQU  C'R'           AGN1 CALL - RETRIEVE OPERATION
000D7    AGNOPOS EQU  C'P'           AGN1 CALL - POSITION OPERATION
000C3    AGNOCURR EQU  C'C'           AGN1 CALL - OCCURRENCE=CURRENT
000C6    AGNOCFIR EQU  C'F'           AGN1 CALL - OCCURRENCE=FIRST
000D3    AGNOCLAS EQU  C'L'           AGN1 CALL - OCCURRENCE=LAST
000E7    AGNOCNEX EQU  C'X'           AGN1 CALL - OCCURRENCE=NEXT
000D7    AGNOCPRI EQU  C'P'           AGN1 CALL - OCCURRENCE=PRIOR
000D5    AGNOCNTH EQU  C'N'           AGN1 CALL - OCCURRENCE=NTH
000D2    AGNOCKEY EQU  C'K'           AGN1 CALL - OCCURRENCE=KEY
000C1    AGNOCALL EQU  C'A'           AGN1 CALL - OCCURRENCE=ALL
000D6    AGNOCOWN EQU  C'O'           AGN1 CALL - OCCURRENCE=OWNER
000E2    AGNOCSTG EQU  C'S'           AGN1 CALL - OCCURRENCE=STG-KEY

```

1.8 #AGNDS

```
COPY #AGNDS
*****
***      AGN:  ADS APPLICATION GENERATOR GENERAL INFORMATION      ***
*** -----
***      AGN IS A DSECT THAT DESCRIBES THE FIELDS IN THE          ***
***      ADSO-APPL-GENERAL RECORD USED BY THE ADS APPLICATION      ***
***      GENERATOR.                                              ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-----------------------------------|-----------------------------------|
| 000000 | AGN DSECT | 09:40:01 10/11/85 |
| 000000 | AGNSEQ DS F | SYSGMT-038 SEQUENCE NUMBER |
| 000004 | AGNMAXRS DS H | MAXIMUM NUMBER OF RESPONSES |
| 000006 | AGNPRTCL DS H | DEFAULT PRINT CLASS |
| 000008 | AGNDMODE DS X | DEFAULT MODE (FAST/STEP) |
| 000C6 | AGNFAST EQU C'F' | FAST MODE IS DEFAULT *MDR85149* |
| 000E2 | AGNSTEP EQU C'S' | STEP MODE IS DEFAULT *MDR85149* |
| 000009 | AGNMDATE DS X | DATE FORMAT FOR MENUS |
| 000C3 | AGNCALN EQU C'C' | CALENDAR DATE *MDR85149* |
| 000C5 | AGNEURO EQU C'E' | EUROPEAN DATE *MDR85149* |
| 000C7 | AGNGREG EQU C'G' | GREGORIAN DATE *MDR85149* |
| 000D1 | AGNJULI EQU C'J' | JULIAN DATE *MDR85149* |
| 00000A | AGNPRDES DS CL8 | DEFAULT PRINT DESTINATION |
| 000012 | AGNDESCR DS CL40 | APPLICATION DESCRIPTION |
| 00003A | AGNENVIR DS X | RUNTIME ENVIRONMENT |
| 000C2 | AGNBTCHEQU C'B' | APPL RUNS BATCH ONLY *MDR85149* |
| 000D6 | AGNONLN EQU C'O' | APPL RUNS ONLINE ONLY *MDR85149* |
| 00003B | * AGNANYM EQU C'A' *DISABLED* | APPL RUNS IN ANY MODE **RQE85275* |
| 000068 | DS 45X | FILLER *MDR85149* |
| 0006C | AGNTYPE DS F | TYPE OF SYSGMT-038 RECORD |
| | AGNLEN EQU *-AGN | LENGTH OF THIS DSECT |

1.9 #AGRDS

```
COPY #AGRDS
*****
***      AGR:  ADS APPLICATION GENERATOR GLOBAL RECORDS
***      ***
***  AGR IS A DSECT THAT DESCRIBES THE FIELDS IN THE
***  ADSO-APPL-GLOBAL-RECS RECORD USED BY THE ADS APPLICATION
***  GENERATOR.
***      ***
*****
```

Offset Value

| | | | |
|--------|---------|-----------|----------------------------|
| 000000 | AGR | DSECT | 11:54:57 04/04/85 |
| 000000 | AGRSEQ | DS F | SYSCMT-038 SEQUENCE NUMBER |
| 000004 | AGRNAME | DS CL32 | GLOBAL RECORD NAME |
| 000024 | AGRVER | DS H | GLOBAL RECORD VERSION |
| 000026 | | DS 66X | FILLER |
| 000068 | AGRTYPE | DS F | TYPE OF SYSCMT-038 RECORD |
| 0006C | AGRLEN | EQU *-AGR | LENGTH OF THIS DSECT |

1.10 #AGWADS

```

COPY #AGWADS
*****
***          GWA : ADS/ONLINE GENERATOR WORK AREA
***          GWA IS A DSECT THAT DEFINES THE
***          ADS GENERATOR. THIS KEPT STORAGE IS MANAGED LIKE THE
***          RECORD BUFFER BLOCKS IN THE ONLINE SYSTEM. SIZE IS GOVERNED
***          BY THE BUFFER SIZES SPECIFIED IN THE OCB. THE FIRST GWA
***          IS ALLOCATED ACCORDING TO OCBPBF SZ, ANY SECONDARY GWA'S
***          ARE THE SIZE SPECIFIED IN OCBSBF SZ. THE GWA CONTAINS ALL
***          THE CONTROL BLOCKS NEEDED TO PROCESS A DIALOG, SYSCTRL, AND
***          MRB'S FOR THE GENERATOR MAPS.
***          -----
***          A SEPARATE CHAIN OF STORAGE AREAS IS MAINTAINED FOR THE USE
***          OF THE PROCESS MODULES. THESE ARE MANAGED EXACTLY AS THE
***          GWA, BUT ARE CALLED PROCESS WORK AREAS, OR "PWA'S". THIS
***          SAME CONTROL BLOCK IS USED TO MANAGE THE PWA STORAGE
***          ROUTINE IN ADSOGNRC.
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------------|--------------------------------|
| 000000 | GWA DSECT | 08:21:53 05/28/86 |
| 000000 | GWAID DS CL4 | "GWA*" OR "PWA*" |
| 000004 | GWANXTA DS A | ADDRESS OF NEXT GWA |
| 000008 | ORG GWANXTA | |
| | GWAPRIM #FLAG X'80' | ON IF PRIMARY GWA |
| 000004 | GWAPRIMI DS 0XL1 | |
| 00080 | GWAPRIMM EQU X'80' | |
| 000004 | DS X | |
| 000005 | ORG | |
| 000008 | GWAPREVA DS A | ADDRESS OF PREVIOUS GWA |
| 00000C | GWASIZE DS F | SIZE OF THIS GWA |
| 000010 | GWAFREEA DS A | ADDRESS OF FIRST FREE SPACE |
| 00014 | GWALEN EQU ((*-GWA+3)/4)*4 | LENGTH OF GWA HEADER PORTION |
| 000000 | GWAWK DSECT | PRESENT IN PRIMARY GWA ONLY |
| 000000 | GWAGFDBA DS A | ADDRESS OF GFDB |
| 000004 | GWAFGRDA DS A | ADDRESS OF FIRST GRD |
| 000008 | GWALGRDA DS A | ADDRESS OF LAST GRD |
| 00000C | GWAGPMEA DS A | ADDRESS OF GPME |
| 000010 | GWAFGRSA DS A | ADDRESS OF FIRST GRSE |
| 000014 | GWALGRSA DS A | ADDRESS OF LAST GRSE |
| 000018 | GWACGRSA DS A | ADDRESS OF CURRENT GRS |
| 00001C | GWAFGSSA DS A | ADDRESS OF FIRST GSS |
| 000020 | GWALGSSA DS A | ADDRESS OF LAST GSS |
| 000024 | GWAFGLEA DS A | ADDRESS OF FIRST GLE IN LIT PL |
| 000028 | GWALGLEA DS A | ADDRESS OF LAST GLE IN LIT PL |
| 00002C | GWAFGLRA DS A | ADDRESS OF FIRST GLR |
| 000030 | GWALGLRA DS A | ADDRESS OF LAST GLR |
| 000034 | GWAFOFDA DS A | ADDRESS OF FIRST OFD |
| 000038 | GWALOFDA DS A | ADDRESS OF LAST OFD |
| 00003C | GWANODE DS CL8 | ALTERNATE DBNAME |
| 000044 | GWADICT DS CL8 | ALTERNATE SUBSCHEMA NAME |
| 00004C | GWAMSGS DS 4F | MESSAGE CODE TABLE |
| 00005C | GWAMSGNM DS H | CURRENT MSG TABLE POSITION |
| 00005E | GWALPOFF DS H | NEXT AVAIL. OFFSET IN LIT POOL |
| 000060 | GWANXTMD DS H | NEXT MODULE TO BE CALLED |
| 000062 | GWANXTFC DS H | FUNCTION CODE FOR NEXT MODULE |
| 000064 | GWARINDX DS H | HIGHEST EXISTING RECORD INDEX |
| 000066 | GWAIRAIX DS H | RECORD INDEX FOR IRA |

| | | | | |
|--------|----------|-------------|-------------|--------------------------------|
| 000068 | GWALITIX | DS | H | RECORD INDEX FOR LITPOOL |
| 00006A | GWAVDBIX | DS | H | RECORD INDEX FOR VDB |
| 00006C | GWAVREIX | DS | H | RECORD INDEX FOR VRE |
| 00006E | GWARSPNM | DS | CL32 | SELECTED RESPONSE NAME |
| 00008E | ORG | | GWARSPNM | |
| 00006E | GWAPGID | DS | 0CL26 | PARM LIST FOR #RTNCB MACRO |
| 00006E | GWAPGNM | DS | CL8 | PROGRAM NAME *RZM86031043 |
| 000076 | GWAPGVER | DS | H | PROGRAM VERSION *RZM86031043 |
| 000078 | GWAPNODE | DS | CL8 | NODE NAME FOR PGM *RZM86031043 |
| 000080 | GWAPDICT | DS | CL8 | DICT NAME FOR PGM *RZM86031043 |
| 000088 | ORG | | | |
| 00008E | GWARCDNM | DS | CL32 | SELECTED RECORD NAME |
| 0000AE | GWAACTN | DS | X | CURRENT ACTION CODE |
| | GWADISP | #FLAG X'80' | | ON IF IN DISPLAY MODE |
| 0000AF | GWADISPI | DS | 0XL1 | |
| | 000080 | GWADISPM | EQU X'80' | |
| | | GWAFAST | #FLAG X'40' | ON IF IN FAST MODE |
| 0000AF | GWAFASTI | DS | 0XL1 | |
| | 000040 | GWAFASTM | EQU X'40' | |
| | | GWAFSEL | #FLAG X'20' | ON IF NEXT FUNCTION SELECTED |
| 0000AF | GWAFSELI | DS | 0XL1 | |
| | 000020 | GWAFSELM | EQU X'20' | |
| | | GWASTMP | #FLAG X'10' | ON WHEN STEP MODE MAPOUT |
| 0000AF | GWASTMPI | DS | 0XL1 | |
| | 000010 | GWASTMPM | EQU X'10' | |
| | | GWAMSG | #FLAG X'08' | ON IF MESSAGES FOR SCREEN |
| 0000AF | GWAMSGI | DS | 0XL1 | |
| | 000008 | GWAMSGM | EQU X'08' | |
| | | GWAMSOV | #FLAG X'04' | ON IF MORE THAN 4 MESSAGES |
| 0000AF | GWAMSOVI | DS | 0XL1 | |
| | 000004 | GWAMSOVM | EQU X'04' | |
| | | GWACERR | #FLAG X'02' | ON WHEN COMPILE ERRORS |
| 0000AF | GWACERRI | DS | 0XL1 | |
| | 000002 | GWACERRM | EQU X'02' | |
| | | GWANDFL | #FLAG X'01' | ON IF DIFFERENT NODE SPECIFIED |
| 0000AF | GWANDFLI | DS | 0XL1 | |
| | 000001 | GWANDFLM | EQU X'01' | |
| 0000AF | GWAFLAG1 | DS | X | |
| | | GWAFIND | #FLAG X'80' | GNDB DOES FIND, NOT OBTAIN |
| 0000B0 | GWAFINDI | DS | 0XL1 | |
| | 000080 | GWAFINDM | EQU X'80' | |
| | | GWAGPM | #FLAG X'40' | ON IF GWACPROA POINTS TO A GPM |
| 0000B0 | GWAGPMI | DS | 0XL1 | |
| | 000040 | GWAGPMM | EQU X'40' | |
| | | GWAPERR | #FLAG X'20' | ERROR FOUND IN LAST NC SCREEN |
| 0000B0 | GWAPERRI | DS | 0XL1 | |
| | 000020 | GWAPERRM | EQU X'20' | |
| | | GWASTSK | #FLAG X'10' | ON IF GEN2 RUNNING AS SUBTASK |
| 0000B0 | GWASTSKI | DS | 0XL1 | |
| | 000010 | GWASTSKM | EQU X'10' | |
| | | GWALR | #FLAG X'08' | ON IF ONLY LR ACCESS ALLOWED |
| 0000B0 | GWALRI | DS | 0XL1 | |
| | 000008 | GWALRM | EQU X'08' | |
| | | GWANOLR | #FLAG X'04' | ON IF NO LR ACCESS ALLOWED |
| 0000B0 | GWANOLRI | DS | 0XL1 | |
| | 000004 | GWANOLRM | EQU X'04' | |
| | | GWANOPR | #FLAG X'02' | ON IF NO PRINT IN BATCH COMPLR |
| 0000B0 | GWANOPRI | DS | 0XL1 | |
| | 000002 | GWANOPRM | EQU X'02' | |
| | | GWABTCH | #FLAG X'01' | ON IF BATCH COMPILER |
| 0000B0 | GWABTCHI | DS | 0XL1 | |
| | 000001 | GWABTCHM | EQU X'01' | |
| 0000B0 | GWAFLAG2 | DS | X | |
| | | GWAYER | #FLAG X'40' | ON IF VERSION QUALIFIERS |
| 0000B1 | GWAYERI | DS | 0XL1 | |
| | 000040 | GWAYERM | EQU X'40' | |

| | | | |
|--------|-----------|---------------------|---------------------------------|
| | GWAALL | #FLAG X'20' | ON IF REGENNING ALL DIALOGS |
| 0000B1 | GWAALLI | DS 0XL1 | |
| 000020 | GWAALLM | EQU X'20' | |
| | GWAMAPR | #FLAG X'10' | ON IF MAP RECORDS PROCESSED |
| 0000B1 | GWAMAPRI | DS 0XL1 | |
| 000010 | GWAMAPRM | EQU X'10' | |
| 0000B1 | GWABFLG1 | DS X | BATCH GENERATOR FLAG BYTE |
| 0000B4 | STMTNUM | DS F | TEXT-088 SEQUENCE NUMBER |
| 0000B8 | INBUF | DS CL80 | INPUT BUFFER AREA |
| 000108 | GWAMSGLN | DS CL4 | RELATIVE LINE NUMS FOR GNRC |
| 00010C | GWAOOTBIX | DS H | RECORD INDEX FOR OTB |
| 00010E | GWAPRBCT | DS H | CURRENT # OF BATCH PRINT BUFFS |
| 000110 | GWAFPRBA | DS A | ADDR OF 1ST BATCH PRINT BUFFER |
| 000114 | GWAFDNBA | DS A | ADDR OF 1ST DIALOG NAME BLOCK |
| 000118 | GWACDNBA | DS A | ADDR OF CURRENT DNB |
| 00011C | GWAFRDBA | DS A | ADDR OF 1ST REQUEST DESC BLK |
| 000120 | GWACRDBA | DS A | ADDR OF CURRENT RDB |
| 000124 | GWAFRVBA | DS A | ADDR OF 1ST REQUEST VER BLOCK |
| 000128 | GWACRVBA | DS A | ADDR OF 1ST REQUEST VER BLOCK |
| 00012C | GWACPRBA | DS A | ADDR OF CURRENT PRINT BUFFER |
| 000130 | GWAPRDBK | DS F | DBK FOR PROG-051 FOR DIALOG |
| 000134 | GWAUSERN | DS CL32 | USER ID FOR USER |
| 000154 | GWAPASSW | DS CL8 | PASSWORD FOR USER |
| 00015C | GWAFPXCA | DS F | 1ST PROCESS HDR EXEC CNTL BLK |
| 000160 | GWALPXCA | DS F | LAST PROC EXECUTABLE CODE BLK |
| 000164 | GWANFDBA | DS F | ADDR OF AREA TO BUILD NEW FDB |
| 000168 | GWANFDBS | DS F | LENGTH OF NEW FDB |
| 00016C | GWACPXCA | DS F | ADDR OF CURRENT PEXC CNTL BLK |
| 000170 | GWACOFTB | DS F | CURRENT ENTRY IN OFFSET TABLE |
| | GWATRUE | #FLAG X'80' | TRUE OFFSET FND IN OFFSET TBL |
| 000174 | GWATRUEI | DS 0XL1 | |
| | 000080 | GWATRUEM EQU X'80' | |
| | | GWAFLSE #FLAG X'40' | FALSE OFFSET FND IN OFFSET TBL |
| 000174 | GWAFLSEI | DS 0XL1 | |
| | 000040 | GWAFLSEM EQU X'40' | |
| | | GWAPRGF #FLAG X'20' | DIALOG FAILED PROG REGIST. |
| 000174 | GWAPRGFI | DS 0XL1 | |
| | 000020 | GWAPRGFM EQU X'20' | |
| | | GWAASEC #FLAG X'10' | ADS PRODUCT SECURITY IS ON |
| 000174 | GWAASECI | DS 0XL1 | |
| | 000010 | GWAASECM EQU X'10' | |
| 000174 | GWACMTST | DS X | FLAG BYTE |
| 000175 | GWAUNXF | DS X | NEXT TASK FLAG (UMBRELLA) |
| 000176 | GWAMSWIX | DS H | REC INDEX FOR MAP SWITCH REC |
| 000178 | GWAUCEA | DS A | ADDR OF SUBTASK PARAMETER LIST |
| 00017C | GWAUDBK | DS A | DBKEY OF USER-047 |
| 000180 | GWANFDBF | DS F | NEW FDB BUFFER SIZE *LCB84117* |
| 000184 | GWAUNXT | DS CL8 | NEXT TASK CODE (UMBRELLA) |
| 00018C | GWARCDVR | DS H | SELECT RECORD VERSION (UMBR) |
| 00018E | GWAPUBL | DS X | PUBL051 VALUE IN PROG-051 |
| | GWAOLM | #FLAG X'80' | ONLINE MAP (GNDB 15) *JMA85078* |
| 00018F | GWAOLMI | DS 0XL1 | |
| | 000080 | GWAOLMM EQU X'80' | |
| | | GWAINPM #FLAG X'40' | INPUT MAP (GNDB 15) *JMA85078* |
| 00018F | GWAINPMI | DS 0XL1 | |
| | 000040 | GWAINPMM EQU X'40' | |
| | | GWAOUTM #FLAG X'20' | OUTPUT MAP (GNDB 15) *JMA85078* |
| 00018F | GWAOUTMI | DS 0XL1 | |
| | 000020 | GWAOUTMM EQU X'20' | |
| | | GWAPRSI #FLAG X'08' | INPUT PARSED |
| 00018F | GWAPRSII | DS 0XL1 | |
| | 000008 | GWAPRSIM EQU X'08' | |
| | | GWAPRSO #FLAG X'04' | OUTPUT PARSED |
| 00018F | GWAPRSOI | DS 0XL1 | |
| | 000004 | GWAPRSOM EQU X'04' | |

| | | | |
|--------|--------------------------|---|--|
| | | | SUSFILE PARSED |
| 00018F | 00002 | GWAPRSS #FLAG X'02' GWAPRSSI DS OXL1 GWAPRSM EQU X'02' | FLAG BYTE *JMA85078* |
| 00018F | | GWAFLAG3 DS X | MESSAGE PARM STACK START ADDR |
| 000190 | | GWASTKST DS A | MESSAGE PARM STACK COUNT |
| 000194 | | GWASTKCT DS F | AN EXTERNAL REC FOR AN*JMA85344* |
| 000198 | 00080 | GWAXIRC #FLAG X'80' GWAXIRCI DS OXL1 GWAXIRCM EQU X'80' | OUTPUT MAP IS ALSO AN *JMA85344* INTERNAL RECORD *JMA85344* AN EXTERNAL REC FOR AN*JMA85344* |
| 000198 | 00040 | * | OUTPUT MAP IS NOT USED*JMA85344* |
| 000198 | 00020 | * | AS AN INTERNAL RECORD *JMA85344* |
| 000198 | | GWACOPY #FLAG X'20' GWACOPYI DS OXL1 GWACOPYM EQU X'20' | COPY FROM SESSION *MET87215* |
| 000198 | 00010 | GWAWNDO #FLAG X'10' | GNRP IN WINDOW MODE *MET87219* |
| 000198 | 00008 | GWAWNDOI DS OXL1 GWAWNDOM EQU X'10' | ALTERNATE WINDOW MAP *MET87243* |
| 000198 | 00004 | GWAMAP2 #FLAG X'08' GWAMAP2I DS OXL1 GWAMAP2M EQU X'08' | PSRC SCRATCH EXISTS *MET87342* |
| 000198 | 00004 | GWAPSRC #FLAG X'04' | WITHIN QUOTED STRING *ECM87272* |
| 000198 | 00002 | GWAPSRCI DS OXL1 GWAPSRCM EQU X'04' | |
| 000198 | 00002 | GWAQUOT #FLAG X'02' | |
| 000198 | 00001 | GWAQUOTI DS OXL1 GWAQUOTM EQU X'02' | |
| 000198 | | GWARSUM #FLAG X'01' | RESUME AFTER OTP CALL *MET87295* |
| 000198 | 00001 | GWARSUMI DS OXL1 GWARSUMM EQU X'01' | |
| 000198 | 00080 | GWAFLAG4 DS X | FLAG BYTE *JMA85344* |
| 000199 | 00080 | GWAGERR #FLAG X'80' | GENERATE ERROR *MET88076* |
| 000199 | 00040 | GWAGERRI DS OXL1 GWAGERRM EQU X'80' | |
| 000199 | 00040 | GWAPEND #FLAG X'40' | DROP PENDING *MET88076* |
| 000199 | 00020 | GWAPENDI DS OXL1 GWAPENDM EQU X'40' | COMMENT STATEMENT GIK91081032 |
| 000199 | 00020 | GWADLM #FLAG X'20' | |
| 00019A | 00001 | GWADLMI DS OXL1 GWADLMM EQU X'20' | |
| 00019A | 00001 | GWAFLAG5 DS X | FLAG BYTE *MET88076* |
| 00019A | 00002 | GWAPROD DS X | PRODUCT VERSION *MET88076* |
| 00019A | 00002 | GWAIDMS EQU 1 | .. IDMS *MET88076* |
| 00019A | 00003 | GWAVSAM EQU 2 | .. VSAM *MET88076* |
| 00019A | 00003 | GWADB2 EQU 3 | .. DB2 *MET88076* |
| 00019B | 00006 | GWAMDTYP DS X | MODULE (AND SRE) TYPE *MET88067* |
| 00019B | 00006 | GWADCTYP EQU 6 | .. DECLARATION *MET88067* |
| 00019B | 00002 | GWAPMTYP EQU 2 | .. PREMAP *MET88067* |
| 00019B | 00003 | GWARSTYP EQU 3 | .. RESPONSE *MET88067* |
| 00019C | | GWAXGRDA DS F | ADDR OF GRD FOR EXTERN*JMA85344* |
| 00019C | * | * | REC FOR OUTPUT MAP *JMA85344* |
| 00019C | * | * | (FOR GNRC & GFDB TO *JMA85344* |
| 00019C | * | * | SET UP INIT COMPRESSED*JMA85344* |
| 00019C | * | * | RECORD) *JMA85344* |
| 0001A0 | GWASIL DS H | LENGTH OF SHIFTIN BZJ | |
| 0001A2 | GWASOL DS H | LENGTH OF SHIFTOUT BZJ | |
| 0001A4 | GWASO DS XL3 | SHIFTOUT SEQUENCE BZJ | |
| 0001A7 | GWASI DS XL3 | SHIFTIN SEQUENCE BZJ | |
| 0001A7 | GWADBCS #FLAG X'40' | ON IF IN DBCS FIELD BZJ | |
| 0001AA | GWADBCSI DS OXL1 | | |
| 0001AA | 00040 GWADBCSM EQU X'40' | | |

| | | |
|--|--------------------------------|--|
| | GWAGLT #FLAG X'80' | ON IF PROCESING G-LITERAL BZJ |
| 0001AA | GWAGLTI DS 0XL1 | |
| 00080 | GWAGLTM EQU X'80' | |
| | GWADBCA #FLAG X'20' | ON IF DBCS SUPPORT ACTIVE BZJ |
| 0001AA | GWADBCAI DS 0XL1 | |
| 00020 | GWADBCAM EQU X'20' | |
| | GWAGLER #FLAG X'10' | G-LITERAL ERROR FLAG *TBL89009* |
| 0001AA | GWAGLERI DS 0XL1 | |
| 00010 | GWAGLERM EQU X'10' | |
| | GWASRRC #FLAG X'08' | On -> Record spec'd for Send/Receive *MCM88071* |
| 0001AA | GWASRRCI DS 0XL1 | |
| 00008 | GWASRRCM EQU X'08' | |
| | * | Off -> Element spec'd for Send/Receive*MCM88071* |
| 0001AA | GWADBCSP DS X | FLAG BYTE FOR DBCS SUPPORT BZJ |
| 0001AB | GWAMLINE DS X | MAP LINE INDEX *MET87244* |
| 0001AC | GWAWIRA DS A | TOOL BOX WIR RCD ADDR *MET87202* |
| 0001B0 | GWASCAIX DS H | SQL COMM AREA INDEX *LMA88030* |
| 0001B2 | GWASCALN DS H | AND LENGTH *LMA88040* |
| 0001B4 | GWAPBFLX DS H | SQL PBUFF INDEX *LMA88030* |
| 0001B6 | GWAPBFLN DS H | AND LENGTH *LMA88040* |
| 0001B8 | GWADBFIX DS H | SQL DBUFF INDEX *LMA88030* |
| 0001BA | GWADBFLN DS H | AND LENGTH *LMA88040* |
| 0001BC | GWQUEID DS 0CL16 | QUEUE ID FOR SUSPEND *MET88064* |
| 0001BC | GWAQDNAM DS CL8 | DIALOG NAME *MET88064* |
| 0001C4 | GWAQDVER DS CL4 | DIALOG VERSION *MET88064* |
| 0001C8 | GWAQTOOL DS CL4 | TOOL ID -- \$DLG *MET88064* |
| 0001CC | GWAGDCLA DS F | ADDRESS OF GDCL *MET88064* |
| 0001D0 | GWAPSRID DS CL8 | PRINT SOURCE SCR ID *MET88064* |
| 0001D8 | GWAFFD@ DS A | Address of first GRD/GRE on chain *MCM88064* |
| | * | to be used to generate FDEs in *MCM88064* |
| | * | the FDB. *MCM88064* |
| 0001DC | GWALFD@ DS A | Address of last GRD/GRE on chain *MCM88064* |
| | * | to be used to generate FDEs in *MCM88064* |
| | * | the FDB. *MCM88064* |
| 0001E0 | GWANFDO DS F | Next offset available in FDE area *MCM88064* |
| | * | *MCM88064* |
| 0001E4 | GWASGRDA DS F | Address of GRD for send/receive arg *MCM88081* |
| 0001E8 | GWASGREA DS F | Address of GRE for send/receive arg *MCM88081* |
| 0001EC | GWACHRL DS H | LENGTH OF A DBCS CHAR *MET89025* |
| 0001EE | DS H | RESERVED *MET89025* |
| 0001F0 | GWAPLIDA DS A | ADDRESS OF 80 BYTE STG PRIMED WITH *JMA91143* |
| | * | PROCESS INFO FOR PRINT BUF MAINTNANCE*JMA91143* |
| 0001F4 | DS 5F | RESERVED *JMA91143* |
| 00208 | GWAWKLEN EQU ((*-GWAWK+3)/4)*4 | LENGTH OF GWA WORK DSECT |
| 00082 | GWAWKLFN EQU ((*-GWAWK+3)/4) | LENGTH OF GWA WORK IN WORDS |
| ***** | | |
| *** THE FOLLOWING FIELDS ARE EXTENSTIONS TO GWAWK FOR ADSA *** | | |
| *** FOR NOW, THIS AREA IS NOT BEING TREATED AS A SEPERATE DSECT. *** | | |
| *** IT SHOULD BE COPIED IMMEDIATELY AFTER #GWADS *** | | |
| *** ----- *** | | |
| *** THE FOLLOWING FIELDS ARE USED TO MAINTAIN 'CURRENCY' FOR *** | | |
| *** ADSA CONTROL BLOCKS *** | | |
| ***** | | |
| 000208 | GWAGABSK DS A | STORAGE KEY OF THE GAB |
| 00020C | GWACRN CY DS 0F | START OF 'CURRENCY' LIST |
| 00020C | GWACGFEK DS F | STORAGE KEY OF CURRENT GFE |
| 000210 | GWACGARK DS F | STORAGE KEY OF CURRENT GAR |
| 000214 | GWACGFRK DS F | STORAGE KEY OF CURRENT GFR |
| 000218 | GWACMSQK DS F | STORAGE KEY OF CURRENT MSQ |
| 00021C | GWACURNK DS F | STORAGE KEY OF CURRENT URN |
| 000220 | GWACGTCK DS F | STORAGE KEY OF CURRENT GTC |
| 000224 | GWACGMEK DS F | STORAGE KEY OF CURRENT GME |
| 000228 | GWACAGLK DS F | STORAGE KEY OF CURRENT AGL |
| 00022C | GWACAGBK DS F | STORAGE KEY OF CURRENT AGB |
| 000230 | GWACARDK DS F | STORAGE KEY OF CURRENT ARD |
| 00028 | GWACRNLN EQU *-GWACRN CY | LENGTH OF 'CURRENCY' LIST |

```
*****
*** THE FOLLOWING FIELDS ARE USED FOR SESSION MANAGEMENT ***
*****
000234 GWAADBTD DS CL8          ADB GENERATE DATE
00023C GWAADBDM DS CL8          ADB GENERATE TIME
000244 GWAQNAME DS OCL16        SESSION QUEUE ID CONSISTING OF:
000244 GWAQATYP DS CL4          QUEUE TYPE ($ADA).      MXN91257
000248 GWAQANAM DS CL8          APPLICATION NAME
000250 GWAQAVER DS CL4          APPLICATION VERSION
000254 GWAQQRA DS A             ADDRESS OF SESSION Q BUFFER
000258 GWAQQRLM DS H            SIZE OF THE SESSION Q BUFFER
                                GWAADB#FLAG X'80'
00025A GWAADBFI DS OXL1         ON = ADB HAS BEEN FOUND
                                00080 GWAADB#FLAG X'80'
                                GWAAREC #FLAG X'40'      ON = SESSION RECOVERY IN PROGRESS
00025A GWAARECI DS OXL1
                                00040 GWAARECM EQU X'40'
                                GWASNQF #FLAG X'20'      ON = VALID SESSION QUEUE EXISTS
00025A GWASNQFI DS OXL1
                                00020 GWASNQFM EQU X'20'
                                GWADACT #FLAG X'10'      ON = DEFINITION IN PROGRESS
00025A GWADACTI DS OXL1
                                00010 GWADACTM EQU X'10'
                                GWATATF #FLAG X'08'      ON = TAT HAS BEEN FOUND
00025A GWATATFI DS OXL1
                                00008 GWATATFM EQU X'08'
                                GWADSPL #FLAG X'04'      ON = CURRENTLY IN DISPLAY MODE
00025A GWADSPLI DS OXL1
                                00004 GWADSPLM EQU X'04'
00025A GWAFLGS DS X             SESSION CONTROL FLAGS
00025B DS X                  RESERVED
*****
*** THESE FIELDS ARE USED BY RELOCATABLE STORAGE MANAGEMENT ***
*****
00025C GWAFLFA DS A             ADDRESS OF FIRST PAGE LIST FRAGMENT
000260 GWALPLFA DS A             ADDRESS OF LAST PAGE LIST FRAGMENT
000264 GWAHIPLF DS H            NUMBER OF PAGE LIST FRAGMENTS
000266 GWAPAGSZ DS H            STORAGE PAGE SIZE IN BYTES
000268 GWANPLFS DS H            NUMBER OF SLOTS IN ONE PLF
00026A DS H                  RESERVED
*****
*** THESE FIELDS ARE MAINTAINED BY ADSOAGN1 AND ADSOADM AND ***
*** USED BY ADSOAGEN TO VALIDATE THE APPLICATION PRIOR TO ***
*** GENERATE ***
*****
00026C GWASFTA DS A             ADDRESS OF SYSTEM FUNCTION TABLE
000270 GWANTSKS DS H            NUMBER OF TASK CODES
000272 GWANAGRC DS H            NUMBER OF APPL GLOBAL RECORDS
000274 GWAURSPS DS H            NUMBER OF UNDEFINED DFLT RESPONSES
000276 GWANRSPS DS H            TOTAL NUMBER OF RESPONSES
000278 GWAUFUNC DS H            NUMBER OF UNDEFINED FUNCTIONS
00027A GWANFUNC DS H            TOTAL NUMBER OF FUNCTIONS
00027A 0009F GWAWAFLN EQU ((*-GWAWK+3)/4) LENGTH OF ADSA GWAWK IN WORDS
00027C GWAWALEN EQU GWAWAFLN*4 LENGTH OF ADSA GWAWK IN BYTES

```

1.11 #AIRDS

```

COPY #AIRDS
*****
*** AIRDS: APPLICATION INTERFACE RECORD DSECT ***
*** ****
***** ****
***** ****



| <u>Offset</u> | <u>Value</u>      |                                    |
|---------------|-------------------|------------------------------------|
| 000000        | AIR DSECT         | 08:21:04 12/03/81                  |
| 000000        | AIRAPPNM DS CL8   | APPLICATION NAME                   |
| 000008        | AIRRESP DS CL8    | MAPPED-IN EXTERNAL FUNCTION        |
| 000010        | AIRDOAID DS CL1   | DC-AID BYTE FROM #TREQ-GET         |
| 000011        | AIRCURIDI DS CL8  | CURRENT DIALOG BEING EXECUTED      |
| 000019        | AIRNXTDI DS CL8   | NEXT DIALOG TO BE INVOKED          |
|               | *                 |                                    |
| 000C3         | AIRCALEN EQU C'C' | CALENDAR DATE FORMAT               |
| 000C5         | AIREUROP EQU C'E' | EUROPEAN DATE FORMAT               |
| 000C7         | AIRGREG EQU C'G'  | GREGORIAN DATE FORMAT              |
| 000D1         | AIRJUL EQU C'J'   | JULIAN DATA FORMAT                 |
| 000021        | AIRDATE DS CL1    | DATE FORMAT                        |
|               | *                 |                                    |
| 000D3         | AIRLOOKU EQU C'L' | USER REQUESTS RESPONSE LOOKUP      |
| 000C9         | AIRINVOK EQU C'I' | USER REQUESTS RESPONSE INVOKE      |
| 000D4         | AIRMENU EQU C'M'  | USER REQUESTS MENU CONSTRUCTION    |
| 000022        | AIRSERVC DS CL1   | USER'S SERVICE REQUEST             |
|               | *                 |                                    |
| 0005C         | AIRMSGPN EQU C'*' | MESSAGE IS PENDING                 |
| 000023        | AIRMSGSW DS CL1   | MESSAGE PENDING INDICATOR SWITCH   |
|               | *                 |                                    |
| 000C2         | AIRBACK EQU C'B'  | PAGE BACKWARD                      |
| 000C6         | AIRFWARD EQU C'F' | PAGE FORWARD                       |
| 000E2         | AIRSAME EQU C'S'  | SAME DIRECTION                     |
| 000D3         | AIRLEFT EQU C'L'  | PAGE LEFT                          |
| 000D9         | AIRRIGHT EQU C'R' | PAGE RIGHT                         |
| 000D7         | AIRPOP EQU C'P'   | POP EXPLOSION LEVEL                |
| 000E3         | AIRTOP EQU C'T'   | POP TO TOP LEVEL                   |
| 000024        | AIRDIREC DS CL1   | PROCESS DIRECTION                  |
|               | *                 |                                    |
| 00000         | AIRRTN00 EQU 0    | VALID FUNCTION AND SECURITY        |
| 00001         | AIRRTN01 EQU 1    | ADDRESS OF AIR INVALID             |
| 00002         | AIRRTN02 EQU 2    | APPLICATION NAME INVALID           |
| 00003         | AIRRTN03 EQU 3    | ADDRESS OF MENU MAP RECORD INVALID |
| 00004         | AIRRTN04 EQU 4    | FDE NOT FOUND FOR CURR FUNCTION    |
| 00005         | AIRRTN05 EQU 5    | CORRUPTED AIRCFDEA OR AIRCFUNC     |
| 00006         | AIRRTN06 EQU 6    | RESPONSE INVALID FOR CURR FUNC     |
| 00007         | AIRRTN07 EQU 7    | PF-KEY INVALID FOR APPL OR FUNC    |
| 00008         | AIRRTN08 EQU 8    | NEXT FUNCTION NOT ADS-DIALOG       |
| 00009         | AIRRTN09 EQU 9    | INVALID SECURITY FOR NEXT DIALOG   |
| 0000A         | AIRRTN10 EQU 10   | RESPONSE IS SYSTEM FUNCTION        |
| 0000B         | AIRRTN11 EQU 11   | RESPONSE IS HELP FUNCTION          |
| 0000C         | AIRRTN12 EQU 12   | INVALID MENU DEFN ELMT ADDRESS     |
| 0000D         | AIRRTN13 EQU 13   | INVALID MENU ALE ADDRESS           |
| 0000E         | AIRRTN14 EQU 14   | INVALID MENU FDE INDEX             |
| 0000F         | AIRRTN15 EQU 15   | RESPONSE INVALID FOR CURR FUNC     |
| 00010         | AIRRTN16 EQU 16   | PF-KEY INVALID FOR APPL OR FUNC    |
| 00011         | AIRRTN17 EQU 17   | NEXT FUNCTION NOT ADS-DIALOG       |
| 00012         | AIRRTN18 EQU 18   | INVALID SECURITY FOR NEXT DIALOG   |
| 00013         | AIRRTN19 EQU 19   | FUNCTION IS A SYSTEM FUNCTION      |
| 00014         | AIRRTN20 EQU 20   | FUNCTION IS THE HELP FUNCTION      |
| 000025        | AIRRTNCD DS CL2   | RETURN STATUS CODE FROM AIR        |
|               | *                 |                                    |


```

| | | | | |
|--------|--------------|-----------------|------|--------------------------------|
| | 000E2 | AIRSTEP EQU | C'S' | DIALOG IS IN STEP MODE |
| | 000C6 | AIRFAST EQU | C'F' | DIALOG IS IN FAST MODE |
| 000027 | | AIRMODE DS | CL4 | DIALOG EXECUTION MODE |
| | | * | | |
| | 00002B | AIRUSER DS | CL32 | DC-USER ID |
| | 00004B | AIRINSTL DS | CL32 | DC-USER INSTALLATION CODE |
| | 00006B | AIRSTMSG DS | CL39 | STANDARD MESSAGE CODE |
| | 000092 | AIRDBUG1 DS | CL25 | USER DEBUG FIELD 1 |
| | 0000AB | AIRDBUG2 DS | CL8 | USER DEBUG FIELD 2 |
| | 0000B3 | DS | CL29 | RESERVED |
| | 0000D0 | AIRDECML DS | CL8 | CVD/UNPACK AREA FOR AIRRTNCD |
| | 0000D8 | AIRCFUNC DS | CL8 | CURRENT FUNCTION'S NAME |
| | 0000E0 | AIRNFUNC DS | CL8 | NEXT FUNCTION'S NAME |
| | 0000E8 | AIRCFDEA DS | A | CURRENT FUNCTION'S FDE ADDRESS |
| | 0000EC | AIRNFDEA DS | A | NEXT FUNCTION'S FDE ADDRESS |
| | 0000F0 | AIRADEA DS | A | APPLICATION'S ADE ADDRESS |
| | 0000F4 | AIRREGSV DS | 16F | REGISTER SAVE AREA (RESERVED) |
| 00134 | AIRDSLEN EQU | ((*-AIR+3)/4)*4 | | LENGTH OF DSECT |

1.12 #AMHDS

```
COPY #AMHDS
*****
***      AMH:  ADS APPLICATION GENERATOR FUNCTION MENU HEADER ***
***      ***
***  AMH IS A DSECT THAT DESCRIBES THE FIELDS IN THE ***
***  ADSO-FUNC-MENU-HDR RECORD USED BY THE ADS APPLICATION ***
***  GENERATOR.                                              ***
***      ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|---------------|
| 000000 | AMH | DSECT |
| 000000 | AMHSEQ | DS F |
| 000004 | AMHLINE# | DS X |
| 000005 | AMHTEXT | DS CL79 |
| 000054 | | DS 20X |
| 000068 | AMHTYPE | DS F |
| 0006C | AMHLEN | EQU *-AMH |
| 0001B | AMHLENF | EQU (*-AMH)/4 |

| | | | |
|--------|----------|---------------|-----------------------------------|
| 000000 | AMH | DSECT | 12:14:37 04/04/85 |
| 000000 | AMHSEQ | DS F | MODCMT-038 SEQUENCE NUMBER |
| 000004 | AMHLINE# | DS X | LINE NUMBER |
| 000005 | AMHTEXT | DS CL79 | MENU TEXT |
| 000054 | | DS 20X | FILLER |
| 000068 | AMHTYPE | DS F | TYPE OF MODCMT-038 RECORD |
| 0006C | AMHLEN | EQU *-AMH | LENGTH OF DSECT |
| 0001B | AMHLENF | EQU (*-AMH)/4 | LENGTH OF DESCRIPTOR IN FULLWORDS |

1.13 #AMRDS

```

COPY #AMRDS
*****
***          AMR: ADSO APPLICATION MENU RECORD
***          ***
***          AMR IS A DSECT THAT DESCRIBES THE ADSO APPLICATION MENU
***          RECORD USED BY THE ADSO RUNTIME SYSTEM. THIS RECORD IS
***          USED WHEN A USER IS IN APPLICATION CONTROL FACILITY(ACF)
***          MODE FOR ACF MENU MAPPING.
***          ***
*****

```

Offset Value

| | | | |
|--------|----------|---|----------------------------------|
| 000000 | AMR | DSECT | 09:18:17 01/28/83 |
| 000000 | AMRPAGE | DS H | PAGE NUMBER |
| 000002 | AMRTOTPG | DS H | TOTAL PAGE NUMBER |
| 000004 | AMRNPAGE | DS H | NEXT PAGE NUMBER |
| 000006 | AMRHDR | DS OCL237 | MAP HEADER |
| 000006 | AMRHDRL1 | DS CL79 | LINE 1 OF THE MAP HEADER |
| 000055 | AMRHDRL2 | DS CL79 | LINE 2 OF THE MAP HEADER |
| 0000A4 | AMRHDRL3 | DS CL79 | LINE 3 OF THE MAP HEADER |
| 0004F | AMRHRLN | EQU *-AMRHDRL3 | LENGTH OF A HEADER LINE |
| 0000F3 | AMRDATE | DS CL8 | CURRENT DATE |
| 0000FB | AMRDIALG | DS CL8 | DIALOG |
| 000103 | AMRRESPF | DS CL8 | RESPONSE FIELD |
| 00010B | AMRMODE | DS CL4 | STEP/FAST MODE |
| 00010F | AMRPASS | DS CL32 | PASSING |
| 00012F | AMRUSRID | DS CL32 | USER ID |
| 00014F | AMRPSWRD | DS CL8 | PASSWORD |
| 00157 | AMRFLEN | EQU *-AMR | LENGTH OF FIXED PORTION OF AMR |
| | | *** SELECTION SECTION, MAY CONTAIN UP TO 50 ENTRIES | |
| 000157 | AMRTABLE | DS 50CL38 | TABLE OF 50 SELECTION ENTRIES |
| 0008C3 | ORG | AMRTABLE | |
| 000157 | AMRSNTRY | DS OCL38 | SELECTION ENTRY |
| 000157 | AMRSLECT | DS CL1 | SELECT FLAG |
| 000158 | AMRSRESP | DS CL8 | SEPERATE RESPONSE FIELD |
| 000160 | AMRSKEY | DS CL1 | KEY ASSOCIATED WITH RESPONSE |
| 000161 | AMRDSCRT | DS CL28 | DESCRIPTION |
| 00026 | AMRSLEN | EQU *-AMRSNTRY | LENGTH OF A SELECT ELEMENT ENTRY |
| 00017D | ORG | | |
| 0076C | AMRTBLEN | EQU *-AMRTABLE | LENGTH OF SELECT ELEMENT TABLE |
| 00032 | AMRSLNUM | EQU AMRTBLEN/AMRSLEN | MAXIMUM NUMBER OF ENTRIES |
| 008C3 | AMRLEN | EQU *-AMR | LENGTH OF ENTIRE AMR |

1.14 #ANCBDS

```

COPY #ANCBDS
*****
***      ANCB: NODE CONTROL BLOCK FOR ADS/ONLINE GENERATOR ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|--------------|--|
| 000000 | NCB | DSECT | 07/06/90 14:24:59 |
| 000000 | NCBTYP | DS CL1 | NODE TYPE CODE |
| 000000 | NCBFLU | EQU X'00' | 00 - FLUSH NODE |
| 000001 | NCBKWD | EQU X'01' | 01 - KEYWORD NODE |
| 000002 | NCBNAM | EQU X'02' | 02 - CHARACTER STRING NODE |
| 000003 | NCBNUM | EQU X'03' | 03 - NUMERIC STRING NODE |
| 000004 | NCBSTR | EQU X'04' | 04 - DELIMITED STRING NODE |
| 000005 | NCBTRM | EQU X'05' | 05 - TERMINAL NODE |
| 000006 | NCBERN | EQU X'06' | 06 - ERROR NODE |
| 000008 | NCBANM | EQU X'08' | 08 - ADS/O DATANAME NODE |
| 000009 | NCBEXP | EQU X'09' | 09 - PHRASE NODE |
| 00000A | NCBDEC | EQU X'0A' | 10 - DECIMAL NUMBER |
| 000017 | NCBBIN | EQU X'17' | 23 - BINARY STRING NODE |
| 000019 | NCBHVR | EQU X'19' | 25 - HOST VARIABLE NODE |
| 000001 | NCBSUCC | DS AL2 | SUCCESS NODE OFFSET |
| 000003 | NCBFAIL | DS AL2 | FAILURE NODE OFFSET |
| 000005 | NCBMIN | DS AL1 | MINIMUM # OF CHARACTERS |
| 000006 | NCBMAX | DS AL1 | MAXIMUM # OF CHARACTERS |
| 000007 | NCBLIT | DS CL1 | LITERAL STRING FOR COMPARAND |
| 000008 | | ORG NCBSUCC | |
| 000001 | NCBNXT | DS AL2 | NEXT NODE OFFSET |
| 000003 | NCBRCH | DS AL2 | OFFSET INTO BRANCH TBL/PARSE TREE |
| 000005 | NCBMODIX | DS AL1 | MODULE INDEX FOR SUBTREE |
| 000006 | | ORG NCBBRCH | |
| 000003 | NCBERR | DS AL2 | MESSAGE DICTIONARY ERROR CODE |
| 000005 | NCBSEV | DS CL1 | SEVERITY LEVEL: W -WARNING E - ERROR F - FATAL |
| 000006 | | ORG NCBMIN | |
| 000005 | NCBBAD | DS AL2 | ERROR NODE OFFSET |
| 000007 | NCBSUB | DS AL1 | EXPR/COND NODE SUBTYPE |
| 000008 | | ORG NCBMODIX | |
| 000005 | NCBBCUP | DS AL2 | BACKUP NODE OFFSET |

1.15 #APGDS

```

COPY #APGDS
*****
***      MACRO #APGDS
***      APG: APPLICATION GLOBAL RECORD TO BE USED FOR THE ADSO
***          APPLICATION CONTROL FACILITY (ACF)
***          DICTIONARY NAME IS ADSO-APPLICATION-GLOBAL-RECORD
***          ****
*****

```

Offset Value

| | | | |
|--------|---------|-------------------|---|
| 000000 | APG | DSECT | 12:46:33 08/13/85 |
| 000000 | APGAPPL | DS CL8 | APPLICATION NAME |
| 000008 | APGFUNC | DS CL8 | CURRENT FUNCTION NAME |
| 000010 | APGNXFN | DS CL8 | NEXT FUNCTION NAME |
| 000018 | APGRESP | DS CL8 | CURRENT RESPONSE NAME |
| 000020 | APGDFRS | DS CL8 | DEFAULT RESPONSE NAME |
| 000028 | APGTASK | DS CL8 | TASK CODE WHICH INVOKED THE APPLICATION |
| 000030 | APGXDLG | DS CL8 | EXIT DIALOG NAME |
| 000038 | APGPDST | DS CL8 | DEFAULT PRINT DESTINATION |
| 000040 | APGDATE | DS CL8 | DATE IN FORMAT REQUIRED |
| 000048 | APGUSID | DS CL32 | SIGNED ON USER ID |
| 000068 | APGSECD | DS CL32 | USER'S SECURITY CODES |
| 000088 | APGINST | DS CL32 | USER'S INSTALLATION SECURITY CODES |
| 0000A8 | APGPASD | DS CL128 | PASSED DATA FIELD |
| 000128 | APGAVER | DS H | APPLICATION VERSION NUMBER |
| 00012A | APGASCL | DS H | APPLICATION SECURITY CLASS |
| 00012C | APGRSCL | DS H | RESPONSE SECURITY CLASS |
| 00012E | APGPRCL | DS H | DEFAULT PRINT CLASS |
| 000130 | APGMODE | DS CL4 | MODE (STEP OR FAST) |
| 000134 | APGDFMT | DS CL1 | DATE FORMAT CODE |
| | 000C3 | APGFMDY EQU C'C' | FORMAT IS MM/DD/YY |
| | 000C5 | APGFDMY EQU C'E' | FORMAT IS DD/MM/YY |
| | 000C7 | APGFYMD EQU C'G' | FORMAT IS YY/MM/DD |
| | 000D1 | APGFYD EQU C'J' | FORMAT IS YY/DDD |
| 000135 | APGAID | DS CL1 | AID BYTE |
| 000136 | APGCTYP | DS CL1 | CURRENT FUNCTION TYPE |
| | 000C4 | APGDIAL EQU C'D' | FUNCTION IS A SIMPLE DIALOG |
| | 000D4 | APGMENU EQU C'M' | FUNCTION IS A MENU/DIALOG |
| | 000E2 | APGSIGN EQU C'S' | FUNCTION IS A SIGNON DIALOG |
| 000137 | APGNTYP | DS CL1 | NEXT FUNCTION TYPE |
| | 000C2 | APGNBWD EQU C'B' | NEXT FUNCTION IS A BACKWARD SYSTEM FUNC |
| | 000C4 | APGNDIAL EQU C'D' | NEXT FUNCTION IS A SIMPLE DIALOG |
| | 000C5 | APGNESCP EQU C'E' | NEXT FUNCTION IS AN ESCAPE SYSTEM FUNC |
| | 000C6 | APGNFWD EQU C'F' | NEXT FUNCTION IS A FORWARD SYSTEM FUNC |
| | 000C7 | APGNMENS EQU C'G' | NEXT FUNCTION IS A SYSTEM MENU |
| | 000C8 | APGNHELP EQU C'H' | NEXT FUNCTION IS A HELP SYSTEM FUNC |
| | 000D4 | APGNMENU EQU C'M' | NEXT FUNCTION IS A MENU/DIALOG |
| | 000D5 | APGNSIGS EQU C'N' | NEXT FUNCTION IS A SIGNON MENU |
| | 000D6 | APGNPOP EQU C'O' | NEXT FUNCTION IS A POP SYSTEM FUNC |
| | 000D7 | APGNPGM EQU C'P' | NEXT FUNCTION IS A USER PROGRAM |
| | 000D8 | APGNQUIT EQU C'Q' | NEXT FUNCTION IS A QUIT SYSTEM FUNC |
| | 000D9 | APGNRET EQU C'R' | NEXT FUNCTION IS A RETURN SYSTEM FUNC |
| | 000E2 | APGNSIGN EQU C'S' | NEXT FUNCTION IS A SIGNON DIALOG |
| | 000E3 | APGNTOP EQU C'T' | NEXT FUNCTION IS A RETURN TO TOP SYSFUN |
| | 000E4 | APGNPOPT EQU C'U' | NEXT FUNCTION IS A POPTOP SYSTEM FUNC |
| | 000E7 | APGNSOSF EQU C'X' | NEXT FUNCTION IS A SIGNON SYSTEM FUNC |
| | 000E8 | APGNSFSF EQU C'Y' | NEXT FUNCTION IS A SIGNOFF SYSTEM FUNC |
| 000138 | APGCTRL | DS CL1 | NEXT FUNCTION CONTROL COMMAND |

| | | | | | |
|--------|-------|----------|-----|-------|---|
| | 000E3 | APGCTR | EQU | C'T' | TRANSFER TO NEXT FUNCTION |
| | 000C9 | APGCINV | EQU | C'I' | INVOKE NEXT FUNCTION |
| | 000D9 | APGCRET | EQU | C'R' | RETURN TO NEXT FUNCTION |
| | 000D3 | APGCLIN | EQU | C'L' | LINK TO DIALOG |
| | 00040 | APGCPGM | EQU | C' ' | LINK TO USER PROGRAM |
| 000139 | | APGSNON | DS | CL1 | SIGNON SWITCH |
| | 000D5 | APGSNO | EQU | C'N' | NO SUCCESSFUL SIGNON |
| | 000E8 | APGSYES | EQU | C'Y' | USER SUCCESSFULLY SIGNED ON |
| 00013A | | APGDGNM | DS | CL8 | DIALOG OR DC PROGRAM NAME |
| 000142 | | APGFNDSC | DS | CL28 | FUNCTION DESCRIPTION |
| 00015E | | APGMSG | DS | CL240 | MESSAGE AREA |
| 00024E | | APGSNRQM | DS | CL1 | SIGNON REQUIREMENTS |
| | 000D9 | APGSREQD | EQU | C'R' | SIGNON REQUIRED |
| | 000D6 | APGSOPTN | EQU | C'O' | SIGNON OPTIONAL |
| | 000D5 | APGSNOT | EQU | C'N' | SIGNON NOT ALLOWED |
| 00024F | | APGMRESP | DS | CL8 | RESPONSE FIELD FOR MAPS |
| 000257 | | | DS | CL54 | RESERVED |
| | 0028D | APGLEN | EQU | *-APG | LENGTH OF THE APPLICATION GLOBAL RECORD |

1.16 #APLGDS

```

COPY #APLGDS
*****
*** #APLGDS: ADS PRINT LOG UTILITY PARAMETERS ***
*** NOTE: PLGLPRMS HAS BEEN PUT INTO A SEPARATE COPY MEMBER ***
*** #PLGLDS FOR USE IN OTHER UTILITIES *PHH86041*
*** ****
***** PRINT NOGEN
*****
*** PLGFPRMS: COMMUNICATION DSECT BETWEEN ADSOPLGP AND ADSOPLGF ***
*** (PARSER AND FORMATTER FOR ADS BATCH PRINT LOG UTILITY) ***
*** ****

Offset Value
000000 PLGFPRMS DSECT 15:48:05 04/29/86
* ----- FULLWORD FIELDS -----
000000 FEYE DS CL4'FWRK'
000004 FEP1R12 DS A ADDRESS OF PLGFEP1 FROM R12
000008 FSUBR12 DS A LEVEL 1 SUBROUTINE BASE
00000C FSUBR2T8 DS 7F LEVEL 1 SUBROUTINE SAVEAREA
000028 FSB2R12 DS A LEVEL 2 SUBROUTINE BASE
00002C FSB2R2T8 DS 7F LEVEL 2 SUBROUTINE SAVEAREA
000048 FPTCHA DS A ADDRESS OF PATCH AREA: PLGPTCH
00004C FPTC2A DS A ADDR OF 2ND PATCHAREA*PHH86119*
000050 FPRTLNA DS A ADDR OF PLGP'S PRINT BUFFER
000054 FSCHKSAV DS F SAVE AREA FOR #ACALL'S STACK
* . CHECKING CODE
000058 FSTACKEA DS A ADDR OF END OF LAST STACK SLOT
00005C ADSCSAA DS A CSA ADDRESS *DKJ85337*
000060 ADSIDMSA DS A IDMS ADDRESS *DKJ85338*
000064 ADSENTP DS 3F PARM LIST *DKJ85338*
000070 FSYSPLST DS 10F FOR #GETSTG & #BRPL PLIST PARM
*
000098 FDOUBLE DS D WORK FIELD FOR EDITING DATE
* . AND TIME (FOR CVD, CVB, AND
* . UNPACK INSTRUCTIONS)
0000A0 FEDDATE DS CL8 EDITED DATE: MM/DD/YY
0000A8 FEDTIME DS CL16 EDITED TIME: HH.MM.SS.TTTT
0000B8 FEDITNUM DS CL16 EDITED NUMBER: 999,999,999
*
0000C8 FLOGBUFA DS F ADDR OF INPUT BUFFER (FOR
* . 'AREA' OF #BRPL)
0000CC FLOGBUFL DS F LENGTH OF INPUT BUFFER (FROM
* . LRECL RETURNED BY #BRPL;
* . THIS MUST BE FWD)
0000D0 FDCMBUFA DS F ADDR OF DECOMPRESS BUFFER
0000D4 FOUTBUFA DS F EITHER FLOGBUFA OR FDCMBUFA
*
* ----- HWD FIELDS FOR PLGF -----
0000D8 FDCMBFH1 DS H LENGTH OF DECOMPRESSION BUFFER
* . (FROM LGSCMPWK IN #LGSDS)
0000DA FDCMRCHL DS H LENGTH OF DECOMPRESSED RECORD
* . (COMPUTED BY DCOMP IN PLGF)
0000DC FLOGRCHL DS H LENGTH OF LOG RECORD (RETURNED
* . BY #BGET, SAVED AS HALFWORD)
0000DE FOCTRCHL DS H LENGTH OF REC TO PRINT (FROM
* . BLPRLEN, OR FDCMRCHL, OR

```

```

* . FROM FLOGRCHL IF NO PREFIX)
*
*
0000E0 FFRODTK DS 0F ----- FWD SYSIN PARMS -----
0000E0 FFRODATE DS F FROM DATE/TIME KEY (CF BLPDTK)
0000E4 FFROTIME DS F PACKED 'FROM' TIME 00YYDDC
BINfmt 'FROM' TIME HHMMSSTTT
. (TIME FUST FOLLOW DATE)

*
0000E8 FTODTK DS 0F TO DATE/TIME KEY (CF. BLPDTK)
0000E8 FTODATE DS F PACKED JULIAN 'TO' TIME YYDDDC
0000EC FTOTIME DS F BINfmt 'TO' TIME (=HHMMSSTTT)
. (TIME FUST FOLLOW DATE)

*
0000F0 FBRPLA DS F ADDRESS OF ACQUIRED BRPL STG
0000F4 FBRPLLEN DS F LENGTH OF BRPL

*
0000F8 FBLKSIZE DS F'0' ----- BLOCK SIZE (REQUIRED FOR DOS)
0000FC FLRECL DS F'0' RECORD SIZE (REQUIRED FOR DOS)

*
*
000100 FFSTPGMB DS A ----- ADDR OF FIRST PGM REQUEST BLK
000104 FCURPGMB DS A ADDR OF CURRENT PGM BLOCK
000108 FFREPGME DS A ADDR OF PGM BLOCK ENTRY
00010C FPGMECNT DS H TOTAL NBR OF PGM ENTRIES USED

*
*
000110 FFSTUSRBLK DS A ----- ADDR OF FIRST USR REQUEST BLK
000114 FCURUSRBLK DS A ADDR OF CURRENT USR BLOCK
000118 FFREUSRE DS A ADDR OF USR BLOCK ENTRY
00011C FUSRECNTR DS H TOTAL NBR OF USR ENTRIES USED

*
*
00011E FDDNAME DS CL8'ADSLOGA' ----- DDNAME OF LOG FILE *PHH86018*
FPGMALL #FLAG X'01' 'PGM ALL' WAS SPECIFIED
FUSRALL #FLAG X'02' 'USR ALL' WAS SPECIFIED
FEXCLUD #FLAG X'08' EXCLUDE WAS SPECIFIED
. (OFF MEANS PRINT)
FNOPRFX #FLAG X'10' LOG FILE HAS NO PREFIX
. (ACCORDING TO SYSIN PARMS)
FMODEEXT #FLAG X'20' MODE OF EXTRACT VS. *PHH86090*
. DEFAULT OF REPORT *PHH86090*
FWRNSTA #FLAG X'40' WARNING ISSUED WHEN *PHH86090*
. MODE=EXTRACT & STATS*PHH86090*
. REC PASSED SELECTION*PHH86090*
000126 FFLAG DS X FLAGBYTE

*
*
*
*
*
THE FEXCLUD FLAG IN FFLAG
. DETERMINES WHETHER THE
. RECORD TYPES BELOW ARE TO
. BE SELECTED OR DESELECTED

*
*
SHOULD HAVE ONE FLAG FOR EACH
. REC TYPE IN #BLPDS EQUATES
. PLUS A FLAG FOR 'ALL';
. FLAGS ARE LISTED IN ORDER
. OF #BLPDS EQUATES.

*
FTYPALL #FLAG X'01' ALL ARE SELECTED/DESELECTED
FTYPDBG #FLAG X'02' DEBUG TYPE " / " *PHH85352*
FTYPUSR #FLAG X'04' USER TYPE " / " *PHH85352*
FTYPSUS #FLAG X'08' SUSPENSE TYPE " / " *PHH85352*
FTYPWTO #FLAG X'10' WTO TYPE " / " *PHH85352*
FTYPABE #FLAG X'20' ABEND TYPE " / " *PHH85352*

```

```

FTYPSTA #FLAG X'40'           STATISTIC TYPE " / " *PHH85352*
FTYPCKP #FLAG X'80'           RESERVED FOR CHECKPT *PHH85352*
*
000127 FTYPFLAG DS   X          PRINT/EXCLUDE TYPES FLAGBYTE
*
000128 FWRKFLAG DS   X          WORK FLAG BYTE FOR ADSOPLGF
*                                         (NOT IN USE CURRENTLY)*PHH86002*
*
00012C FTRCR8  DS   F
000130 FTRCR12 DS   F
000134 FTRCBUF DS   CL80
*
00184 FPRMLEN EQU  *-PLGPRMS
*****
*** PLGFPGMB: BLOCK OF PROGRAM REQUESTS CHAINED OFF OF PLGPRMS ***
*** PLGFPGMB DSECT
000000 PGMBNXTB DS   A          ADDR OF NEXT PLGFPGMB
000004 PGMBPRIB DS   A          ADDR OF PRIOR PLGFPGMB
000008 PGMBSLTF DS   CL8        1ST PGM NAME ENTRY
0000A PGMBSLTN EQU  10         NUMBER OF ENTRIES PER BLOCK
00058 PGMBLEN EQU  PGMBSLTF+(8*PGMBSLTN)-PLGFPGMB
00050 PGMBSLTM EQU  PLGFPGMB+PGMBLEN-8    LAST PGM NAME ENTRY (MAX)
*****
*** PLGFUSR: BLOCK OF PROGRAM REQUESTS CHAINED OFF OF PLGPRMS ***
*** PLGFUSR DSECT
000000 USRBNXTB DS   A          ADDR OF NEXT PLGFUSR
000004 USRBPRIB DS   A          ADDR OF PRIOR PLGFUSR
000008 USRBSLTF DS   CL32      1ST USR NAME ENTRY
00005 USRBSLTN EQU  5          NUMBER OF ENTRIES PER BLOCK
000A8 USRBLEN EQU  USRBSLTF+(32*USRBSLTN)-PLGFUSR
00088 USRBSLTM EQU  PLGFUSR+USRBLN-32    LAST USR NAME ENTRY (MAX)
*
PRINT GEN

```

1.17 #APREDS

```

COPY #APREDS
*****
*** APRE IS THE DSECT THAT DEFINES THE APPLICATION RESPONSE ***
*** ELEMENT WHICH REPRESENTS A RESPONSE IN AN ADS APPLICATION. ***
*****
*****
```

| <u>Offset</u> | <u>Value</u> | | | |
|---------------|--|-------------|----------------------------------|------------|
| 000000 | APRE | DSECT | 10:16:53 10/24/85 | |
| 000000 | APRNAME | DS CL8 | RESPONSE NAME | |
| 000008 | APRDLFE | DS F | DESCRIPTION LITPOOL FINDER | |
| 00000C | | ORG APRDLFE | | |
| 000008 | APRDLLEN | DS X | DESCRIPTION LENGTH | |
| 000009 | APRDLLOF | DS XL3 | LITPOOL OFFSET TO DESCRIPTION | |
| 00000C | APRFNIX | DS H | FDE INDEX OF INVOKED FUNCTION | |
| 00000E | | ORG APRFNIX | | |
| | * | | | |
| | ** FOLLOWING EQUATES ARE FOR SYSTEM FUNCTION CODES | | | |
| 00001 | APRFWD | EQU 1 | PAGE FORWARD | |
| 00002 | APRBWD | EQU 2 | PAGE BACKWARD | |
| 00003 | APRPOP | EQU 3 | POP UP THE MENU STACK | |
| 00004 | APRPTOP | EQU 4 | POP TO TOP OF MENU STACK | |
| 00005 | APRRTN | EQU 5 | RETURN TO NEXT HIGHER FUNCTION | |
| 00006 | APRRTOP | EQU 6 | RETURN TO TOP FUNCTION | |
| 00007 | APRHELP | EQU 7 | HELP | |
| 00008 | APRSON | EQU 8 | SIGNON | |
| 00009 | APRSOFF | EQU 9 | SIGNOFF | |
| 0000A | APRQUIT | EQU 10 | QUIT | |
| 0000B | APRESCP | EQU 11 | ESCAPE | *JMA85225* |
| | * | | | |
| 00000C | APRSFNC | DS H | SYSTEM FUNCTION CODE | |
| 00000E | APRSECL | DS X | SECURITY CLASS | |
| | PRINT NOGEN | | | |
| | APRGLBL #FLAG X'80' | | ON = GLOBAL RESPONSE | |
| | APRSYSF #FLAG X'40' | | ON = SYSTEM FUNCTION INVOKED | |
| | APRINTL #FLAG X'10' | | ON = INTERNAL FUNCTION INVOKED | |
| | APRINVK #FLAG X'08' | | ON IF CONTROL COMMAND = INVOKE | |
| | APRTRAN #FLAG X'04' | | ON IF CONTROL COMMAND = TRANSFER | |
| | APPRETN #FLAG X'02' | | ON IF CONTROL COMMAND = RETURN | |
| | APRLINK #FLAG X'01' | | ON IF CONTROL COMMAND = LINK | |
| 00000F | APRFLAG | DS X | RESPONSE FLAG BYTE | |
| | *** | | | |
| | *** NOTE -- NEXT FLAG BYTE IS MOVED EN MASSE TO AND FROM | | *RQE85267* | |
| | *** THE GAR | | *RQE85267* | |
| | APRCLR #FLAG X'80' | | ON IF RETURN CLEAR | *RQE85254* |
| | APRCNT #FLAG X'40' | | ON IF RETURN CONTINUE | *RQE85254* |
| | APRNOSV #FLAG X'20' | | ON = NOSAVE | *RQE85267* |
| | APRNOFI #FLAG X'10' | | ON = NOFINISH | *RQE85267* |
| | APRIMM #FLAG X'08' | | ON = EXECUTE RESP IMMEDIATELY | *RQE85267* |
| | APRDEFR #FLAG X'04' | | ON = EXECUTE RESP DEFERRED | *RQE85288* |
| 000010 | APRFLG2 | DS X | 2ND FLAG BYTE | *RQE85254* |
| | PRINT GEN | | | |
| 000011 | | DS XL3 | UNUSED | *RQE85254* |
| 000014 | | DS F | UNUSED | *RQE85254* |
| 00018 | APRLEN | EQU *-APRE | LENGTH OF RESPONSE ELEMENT | |

1.18 #ARDDS

```

COPY #ARDDS
*****
***   ARD: (APPLICATION RECORD DEFINITION)
***   ARD IS A DSECT THAT CONTAINS APPLICATION GLOBAL RECORD NAMES ***
***   AND VERSION NUMBERS. THERE IS ONE ARD CONTROL BLOCK FOR EACH ***
***   APPLICATION GLOBAL RECORD. THE ARD'S ARE CHAINED TOGETHER ***
***   WITH NEXT AND PRIOR POINTERS AND ANCHORED IN THE GAB.      ***
***   ****
*****
```

Offset Value

| | | | |
|--------|----------|-----------------|-----------------------------------|
| 000000 | ARD | DSECT | 12:56:48 04/04/85 |
| 000000 | ARDNXTA | DS A | ADDRESS OF NEXT ARD |
| 000004 | ARDPREVA | DS A | ADDRESS OF PREVIOUS ARD |
| 000008 | ARDSEQN | DS F | ARD SEQUENCE NUMBER |
| 00000C | ARDRECN | DS CL32 | APPLICATION GLOBAL RECORD NAME |
| 00002C | ARDRECV | DS H | APPLICATION GLOBAL RECORD VERSION |
| 0002E | ARDLEN | EQU *-ARD | LENGTH OF ARD |
| 0000C | ARDLENF | EQU (*-ARD+3)/4 | LENGTH OF ARD IN FULLWORDS |

1.19 #ARSDS

```
COPY #ARSDS
*****
***      ARS:  ADS APPLICATION GENERATOR RESPONSE SEQUENCE REC ***
***      ARS IS A DSECT THAT DESCRIBES THE FIELDS IN THE ***
***      ADSO-RESP-SEQ RECORD USED BY THE ADS APPLICATION ***
***      GENERATOR.  THIS RECORD IS CURRENTLY STORED AS A ***
***      MODCMT-084 RECORD (TYPE -22). ***
*****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------|-------------------------------|
| 000000 | ARS DSECT | 18:10:16 02/14/83 |
| 000000 | ARSSEQ DS F | MODCMT-084 SEQUENCE NUMBER |
| 000004 | ARSRECN DS H | RECORD NUMBER |
| 000006 | ARSCOUNT DS H | # OF RESPONSES IN THIS RECORD |
| 000008 | ARSRESPS DS 12CL8 | ORDERED RESPONSES FOR MENU |
| 000068 | ARSTYPE DS F | TYPE OF MODCMT-084 RECORD |

1.20 #ASBDS

```

COPY #ASBDS
*****
***          ASB: ADS STATISTICS BLOCK
***          ***
***          ***
***          ASB IS A DSECT THAT DESCRIBES THE ADS STATISTICS BLOCK ***
***          WRITTEN TO THE LOG WHICH CONTAINS THE STATISTICS ***
***          ACCUMULATIONS FOR ADS DIALOGS. THE ASB CONTAINS THE ***
***          SAME COUNTERS AS ARE FOUND IN THE ADS STATISTICS ***
***          ACCUMULATION BLOCK (ASAB) WITH IDENTIFICATION FIELDS ***
***          PLACED IN FRONT.
***          ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|--|-----------------------------------|
| 000000 | ASB | DSECT PRINT NOGEN | |
| | | 11:17:16 08/01/86 | |
| 000000 | ASBTYP | EQU *--ASBTYP | |
| | | *** BEGINNING OF DC LOG PORTION OF ASB | |
| 000000 | ASBTYP | DS X | TYPE FOR DC LOG USE |
| 000001 | | DS X | RESERVED FOR DC LOG USE |
| 000002 | ASBLEN | DS H | LENGTH FOR DC LOG USE |
| 000004 | | DS F | RESERVED FOR DC LOG USE |
| 000008 | ASBDCLEN | EQU *--ASBDCLEN | LENGTH OF DC LOG PREFIX |
| | | *** BEGINNING OF ADS PORTION OF ASB | |
| 000008 | ASBID | DS OCL40 | ADS STATS BLOCK ID |
| 000008 | ASBUSID | DS CL32 | DC USER ID |
| 000028 | ASBLTEID | DS CL8 | DC LTERM ID |
| 000030 | ASBUUD | DS CL8 | DIALOG NAME OR "\$ADS@AO" |
| 000038 | ASBUDAT | DS PL4 | DATE TSB BIND COMMAND ISSUED |
| 00003C | ASBUTIM | DS F | TIME TSB BIND COMMAND ISSUED |
| 000040 | | DS 2F | RESERVED |
| 000048 | ASBDSLLEN | EQU *--ASBDSLLEN | LENGTH OF ASB HEADER |
| | | ***** | |
| | | *** ASAB: ADS STATISTICS ACCUMULATION BLOCK | |
| | | *** *** | |
| | | *** ASAB IS A DSECT THAT DESCRIBES THE ADS STATISTICS | |
| | | *** ACCUMULATION BLOCK USED TO HOLD ADS STATISTICS FOR A | |
| | | *** DIALOG. | |
| | | *** *** | |
| 000048 | ASAB | DS OF | ***** |
| 000048 | ASADVER | DS H | DIALOG VERSION NUMBER |
| | ASAINTA | #FLAG X'80' | ON IF INITIAL ACCEPT DONE FOR TSB |
| | ASALNKL | #FLAG X'40' | ON IF INITIAL LINK LEVEL SET |
| | ASARBBA | #FLAG X'20' | ON IF INITIAL RBB ANALYSIS DONE |
| 00004A | ASAFLAG1 | DS X | ASAB FLAG BYTE |
| 00004B | | DS X | RESERVED |
| | | ***** | |
| | | ** ** | |
| | | ** THIS SECTION HOLDS CONTROL COMMAND COUNTERS FOR EXPLICITLY | |
| | | ** CODED CONTROL COMMANDS | |
| | | ** ** | |
| 00004C | ASADS | DS F | NUMBER OF DISPLAY COMMANDS |
| 000050 | ASADSC | DS F | " DISPLAY CONTINUE |
| 000054 | ASAIN | DS F | " INVOKE |
| 000058 | ASALND | DS F | " LINK TO DIALOG |
| 00005C | ASALNP | DS F | " LINK TO PROGRAM |
| 000060 | ASART | DS F | " RETURN |
| 000064 | ASARTC | DS F | " RETURN CONTINUE |

```

000068      ASATF    DS   F      "      TRANSFER
00006C      ASALVD   DS   F      "      LEAVE ADS
000070      ASALVP   DS   F      "      LEAVE APPLICATION
000074      ASAAB    DS   F      "      ABORT
*****
**          THIS SECTION HOLDS CONTROL COMMAND COUNTERS FOR IMPLICIT ADS  **
**          GENERATED CONTROL COMMANDS                                     **
**          **

*****  

000078      ASAIDS   DS   F      NUMBER OF IMPLICIT DISPLAY COMMANDS
00007C      ASAIIIN  DS   F      "      IMPLICIT INVOKE
000080      ASAILND  DS   F      "      IMPLICIT LINK TO DIALOG
000084      ASAILNP  DS   F      "      IMPLICIT LINK TO PROGRAM
000088      ASAIRT   DS   F      "      IMPLICIT RETURN
00008C      ASAIRTC  DS   F      "      IMPLICIT RETURN CONTINUE
000090      ASAITF   DS   F      "      IMPLICIT TRANSFER
000094      ASAILVD  DS   F      "      IMPLICIT LEAVE ADS
000098      ASAILVP  DS   F      "      IMPLICIT LEAVE APPLICATION
00009C      ASAIAIB  DS   F      "      IMPLICIT ABORT
*****
**          THIS SECTION HOLDS GENERAL DIALOG EXECUTION STATISTICS      **
**          **

*****  

0000A0      ASAPMAP   DS   F      NUMBER OF PREMAP PROCESS EXECUTIONS
0000A4      ASARESP   DS   F      "      RESPONSE PROCESS EXECUTIONS
0000A8      ASASTCL   DS   F      "      STATS ACCUMULATION CALLS
0000AC      ASAESCG   DS   F      "      EXPLICIT SCRATCH GETS
0000B0      ASAESCP   DS   F      "      EXPLICIT SCRATCH PUTS
0000B4      ASAESCD   DS   F      "      EXPLICIT SCRATCH DELETES
0000B8      ASAWRPR   DS   F      "      WRITE TO PRINTER COMMANDS
0000BC      ASAPDN   DS   F      "      PUT NEW DETAIL COMMANDS *JEB84060*
0000C0      ASAPDC   DS   F      "      PUT CURRENT DETAIL CMDS *JEB84060*
0000C4      ASAGD    DS   F      "      GET DETAIL COMMANDS *JEB84060*
0000C8      ASAFCBSZ  DS   F      SIZE OF THE FDB
0000CC      ASAFCBSZ DS   F      SIZE OF THE VDB
0000D0      ASAHILEV  DS   H      HIGHEST LINK LEVEL AT WHICH DIALOG EXECUTED
0000D2      ASAHLLEV  DS   H      LOWEST LINK LEVEL AT WHICH DIALOG EXECUTED
*****
**          THIS SECTION HOLDS RBB USAGE STATISTICS                      **
**          **

*****  

0000D4      ASARBBSC DS   F      NUMBER OF TIMES RBB'S PUT TO SCRATCH
0000D8      ASAMSTG   DS   F      MOST RBB STORAGE USED (ALL DIALOGS)
0000DC      ASAMSFR   DS   F      ... RBB FREE SPACE WHEN MOST STORAGE USED
0000E0      ASALSTG   DS   F      LEAST RBB STORAGE USED (ALL DIALOGS)
0000E4      ASALSFR   DS   F      ... RBB FREE SPACE WHEN LEAST STORAGE USED
0000E8      ASAMSBUF  DS   F      MOST RBB SPACE ACQUIRED FOR THIS DIALOG
0000EC      ASALSBUF  DS   F      LEAST RBB SPACE ACQUIRED FOR THIS DIALOG
0000F0      ASAHLRBB DS   H      HIGHEST NUMBER OF RBB'S USED
0000F2      ASALORBB DS   H      LOWEST NUMBER OF RBB'S USED
PRINT GEN
000AC  ASABDSLN EQU  *-ASAB           LENGTH OF ASAB
000F4  ASRECLEN EQU  *-ASB            LENGTH OF ADS STATS RECORD

```

1.21 #ASCD\\$

```
COPY #ASCD$  
*****  
***  
***      ASC:  ADS APPLICATION GENERATOR SECURITY INFORMATION ***  
***  
***  ASC IS A DSECT THAT DESCRIBES THE FIELDS IN THE ***  
***  ADSO-APPL-SECURITY RECORD USED BY THE ADS APPLICATION ***  
***  GENERATOR.  THIS RECORD IS CURRENTLY STORED AS A ***  
***  SYSCMT-038 RECORD (TYPE -15). ***  
***  
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--------|
| 000000 | ASC | DSECT |
| 000000 | ASCSEQ | DS F |
| 000004 | ASCLASS | DS H |
| 000006 | ASCMENUS | DS X |
| 000007 | ASCSIGN | DS X |
| 000008 | ASCFCNM | DS CL8 |
| 000010 | | DS 88X |
| 000068 | ASCTYPE | DS F |

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|--------|----------------------------|
| 000000 | ASC | DSECT | 11:32:44 10/05/82 |
| 000000 | ASCSEQ | DS F | SYSCMT-038 SEQUENCE NUMBER |
| 000004 | ASCLASS | DS H | SECURITY CLASS |
| 000006 | ASCMENUS | DS X | TYPE OF MENU FLAG |
| 000007 | ASCSIGN | DS X | SIGNON REQUIREMENTS FLAG |
| 000008 | ASCFCNM | DS CL8 | SIGNON FUNCTION NAME |
| 000010 | | DS 88X | FILLER |
| 000068 | ASCTYPE | DS F | TYPE OF SYSCMT-038 RECORD |

1.22 #ASPDS

```
COPY #ASPDS
*****
***          ASP: ADS STATISTICS POOL
***          ***
***          ***
***          ***          ASP IS A DSECT THAT DESCRIBES THE ADS STATISTICS POOL
***          USED TO HOLD ADS STATISTICS ACCUM BLOCKS (ASAB) WHERE
***          THE ADS RUNTIME SYSTEM COLLECTS DIALOG LEVEL STATISTICS.
***          THE ASP IS COMPRISED OF THE HEADER PORTION DESCRIBED
***          HERE AND ONE TSB AND ONE ASAB FOR EACH DIALOG WHICH HAS
***          EXECUTED FOR THE CURRENT APPLICATION.
***          ***
*****
```

| <u>Offset</u> | <u>Value</u> |
|---------------|--|
| 000000 | ASP DSECT 11:54:56 12/29/83 |
| | PRINT NOGEN |
| 000000 | ASPID DS CL4 'ASP*' *** |
| 000004 | ASPNXTA DS F NEXT ASP ADDRESS |
| 000008 | ASPREVA DS F PREVIOUS ASP ADDRESS |
| 00000C | ASPINDEX DS H INDEX OF THIS ASP |
| 00000E | ASPMAX DS H MAX NUMBER OF TSB/ASAB'S PER ASP |
| 000010 | ASPNASAB DS H NUMBER OF TSB/ASAB'S IN THIS ASP |
| 000012 | ASPASASZ DS H SIZE OF ONE TSB/ASAB |
| 00014 | ASPDSEN EQU **-ASP LENGTH OF ASP HEADER |
| | ***** |
| | *** FOLLOWING THE ASP HEADER ARE THE ACCUMULATION BLOCKS, *** ONE TSB AND ONE ASAB FOR EACH DIALOG AND POSSIBLY ONE *** OF EACH FOR "ALL OTHER" DIALOGS, THAT IS, ALL DIALOGS *** FOR WHICH STATISTICS ARE NOT SPECIFICALLY REQUESTED. THE *** TSB COMES FIRST AND IS USED TO ACCUMULATE DB AND DC *** STATS. THEN COMES THE ASAB CONTAINING STATS WHICH ARE *** PECULIAR TO ADS DIALOGS. THE ASAB DSECT IS FOUND IN THE *** #ASBDS COPY MEMBER. |
| | *** *** ***** |
| | COPY #TSBDS |
| | ***** |
| | *** TSB: TRANSACTION STATISTICS BLOCK |
| | *** *** *** POINTED TO BY LTETSBA |
| | *** *** *** NOTE: ANY CHANGES IN THIS DSECT SHOULD BE ACCCOMPANIED BY *** SIMILAR CHANGES IN #STRDS AND #TSTDSD. |
| | *** *** ***** |
| 000000 | TSB DSECT 06/07/90 15:27:42 09/06/90 |
| | * |
| | * DC STATISTICS |
| | * |
| 000000 | DS 2F FOR SYSTEM INTERNAL USE ONLY! |
| 000008 | TSBDC DS 0F START OF DC STATISTICS. |
| 000008 | TSBPGMCL DS F NUMBER OF PROGRAMS CALLED. |
| 00000C | TSBPGMLD DS F NUMBER OF PROGRAMS LOADED. |
| 000010 | TSBTRMRD DS F NUMBER OF TERMINAL READS. |
| 000014 | TSBTRMWR DS F NUMBER OF TERMINAL WRITES. |
| 000018 | TSBTRMER DS F NUMBER OF TERMINAL ERRORS. |
| 00001C | TSBSTGGT DS F NUMBER OF STORAGE ACQUISITIONS. |
| 000020 | TSBSCRGT DS F NUMBER OF SCRATCH GETS. |

| | | | |
|---------------------|--------------|------------|-------------------------------------|
| 000024 | TSBSCRPT DS | F | NUMBER OF SCRATCH PUTS. |
| 000028 | TSBSCRDL DS | F | NUMBER OF SCRATCH DELETES. |
| 00002C | TSBQUEGT DS | F | NUMBER OF QUEUE GETS. |
| 000030 | TSBQUEPT DS | F | NUMBER OF QUEUE PUTS. |
| 000034 | TSBQUEDL DS | F | NUMBER OF QUEUE DELETES. |
| 000038 | TSBGETIM DS | F | NUMBER OF GETTIME REQUESTS. |
| 00003C | TSBSETIM DS | F | NUMBER OF SETTIME REQUESTS. |
| 000040 | TSBDBSVR DS | F | NUMBER OF DB SERVICE RQSTS 9/87 |
| 000044 | TSBHISTK DS | F | MAX WORDS USED IN STACK. |
| 000048 | TSBTIMUS DS | F | USER MODE TIME (10**-4 SECONDS). |
| 00004C | TSBTIMSY DS | F | SYSTEM MODE TIME ("). |
| 000050 | TSBTIMWT DS | F | WAIT TIME (10**-4 SECONDS). |
| 000054 | TSBHIRCE DS | F | MAXIMUM NBR OF RCE'S USED |
| 000058 | TSBHIRLE DS | F | MAXIMUM NBR OF RLE'S USED |
| 00005C | TSBHIDPE DS | F | MAXIMUM NBR OF DPE'S USED |
| 000060 | TSBSTDHW DS | F | STORAGE HIGH WATER MARK |
| 000064 | TSBSTDGFR DS | F | NUMBER OF FREESTG REQUESTS |
| 000068 | TSBSVRQS DS | F | NUMBER OF SYS SERVICE REQUESTS |
| 00006C | TSBDCEND DS | 0F | END OF DC STAT PORTION. |
| * | | | |
| * DB STATISTICS | | | |
| * | | | |
| 00006C | | DS 2F | FOR SYSTEM INTERNAL USE ONLY! |
| 000074 | TSBDB DS | 0F | START OF DB STATISTICS. |
| 000074 | TSBPAGRDS | F | NUMBER OF PAGES READ. |
| 000078 | TSBPAGWR DS | F | NUMBER OF PAGES WRITTEN. |
| 00007C | TSBPAGRQ DS | F | NUMBER OF PAGES REQUESTED. |
| 000080 | TSBCALNO DS | F | NUMBER OF CALC RECS WITH NO OFLOW. |
| 000084 | TSBCALOF DS | F | NUMBER OF CALC RECS WITH OFLOW. |
| 000088 | TSBVIANO DS | F | NUMBER OF VIA RECS WITH NO OFLOW. |
| 00008C | TSBVIAOF DS | F | NUMBER OF VIA RECS WITH OFLOW. |
| 000090 | TSBRECRQ DS | F | NUMBER OF RECORDS REQUESTED. |
| 000094 | TSBRECCU DS | F | NUMBER OF RECS CURRENT OF R-U. |
| 000098 | TSBDBRQS DS | F | NUMBER OF DBMS CALLS 9/87 |
| 00009C | TSBFRAGS DS | F | NUMBER OF FRAGMENTS STORED. |
| 0000A0 | TSBRELO DS | F | NUMBER OF RECORDS RELOCATED. |
| 0000A4 | TSBTLOCK DS | F | TOTAL no. locks acquired JDM 5/90 |
| 0000A8 | TSBSLOCK DS | F | Total share locks held JDM 5/90 |
| 0000AC | TSBULOCK DS | F | Total nonshare locks held JDM 5/90 |
| 0000B0 | TSBTLKF DS | F | Total locks freed JDM 5/90 |
| 0000B4 | TSBDBEND DS | 0F | END OF DB STAT PORTION. 10/83 |
| * | | | |
| 0000B4 | TSBIXSTA DS | 0F | START OF INDEX STATISTICS JDM 5/90 |
| 0000B4 | TSB#SPLT DS | F | # OF SR8 SPLITS JDM 5/90 |
| 0000B8 | TSB#SPWN DS | F | # OF SR8 SPAWNS JDM 5/90 |
| 0000BC | TSB#SR8S DS | F | # OF SR8S STORED JDM 5/90 |
| 0000C0 | TSB#SR8D DS | F | # OF SR8S ERASED JDM 5/90 |
| 0000C4 | TSB#SR7S DS | F | # OF SR7S STORED JDM 5/90 |
| 0000C8 | TSB#SR7D DS | F | # OF SR7S ERASED JDM 5/90 |
| 0000CC | TSB#BISR DS | F | # OF B-TREE SEARCHES JDM 5/90 |
| 0000D0 | TSB#LEVEL DS | F | # OF B-TREE LEVELS SRCHED JDM 5/90 |
| 0000D4 | TSB#ORPH DS | F | # OF ORPHANS ADOPTED JDM 5/90 |
| 0000D8 | TSBBISRL DS | H | # OF LVLS SRCHED-BEST CASE 5/90 |
| 0000DA | TSBBISRH DS | H | # OF LVLS SRCHED-WORST CASE 5/90 |
| 00028 | TSBIXSTL EQU | *-TSBIXSTA | LENGTH OF INDEX STATISTICS JDM 5/90 |
| * | | | |
| * Extended DB Stats | | | |
| * | | | |
| 0000DC | TSBXDSTA DS | 0F | Start of Misc DB Stats. LRD95310 |
| 0000DC | TSBUPCNT DS | F | Records Updated. LRD95310 |
| 0000E0 | TSBCACHE DS | F | Pages Found In Cache. LRD95310 |
| 0000E4 | TSBPRFET DS | F | Pg Found in Prefetch Buff LRD95310 |
| 0000E8 | DS 2F | | Future Expansion. LRD95310 |
| * | | | |
| 0000F0 | TSBESTAT DS | 0C | END OF STATISTICS AREA 10/83 |
| * | | | |

```

* HEADER INFORMATION
*
0000F0      DS   2F          FOR SYSTEM INTERNAL USE ONLY!
0000F8      TSBID DS  OCL40  STATISTICS BLOCK ID.
0000F8      TSBUSSID DS CL32  DC USER ID.
000118      TSBLETEID DS CL8   DC LTERM ID.
000120      TSBUID DS  CL8   USER SUPPLIED IDENTIFIER 10/83
000128      TSBUDAT DS  PL4   DATE BIND COMMAND ISSUED 10/83
00012C      TSBUТИM DS  F    TIME BIND COMMAND ISSUED 10/83
                  PRINT NOGEN
000130      TSBFFLAGS DS  0F   10/83
                  TSBTSBA #FLAG X'04' OPT - TSBA= SPECIFIED 10/83
                  TSBACUM #FLAG X'02' OPT - ACCEPT/ACCUMULATE 10/83
                  TSBBDNT #FLAG X'01' OPT - BIND TO TASK START 10/83
000130      TSBFGLG0 DS  X   #TRNSTAT FLAG BYTE 0 (OPTS) 10/83
                  TSBSTRT #FLAG X'01' BIND REQUEST 10/83
                  TSBGET  #FLAG X'02' ACCEPT REQUEST 10/83
                  TSBEND  #FLAG X'04' END REQUEST 10/83
000131      TSBFGLG1 DS  X   #TRNSTAT FLAG BYTE 1 (TYPE) 10/83
                  TSBTASK #FLAG X'01' STATS RELATIVE TO TASK START 10/83
000132      TSBFGLG2 DS  X   #TRNSTAT FLAG BYTE 2 (MODS) 10/83
000133      TSBFGLG3 DS  X   RESERVED 10/83
*
* SQL STATISTICS
*
000134      DS   2F          RESERVED FOR SYSTEM USE JDM 8/90
00013C      TSBSQL DS  0F   Start of SQL statistics JDM 8/90
00013C      TSB#CMD DS  F   ..# SQL commands executed JDM 8/90
000140      TSB#FET DS  F   ..# rows fetched JDM 8/90
000144      TSB#INS DS  F   ..# rows inserted JDM 8/90
000148      TSB#UPD DS  F   ..# rows updated JDM 8/90
00014C      TSB#DEL DS  F   ..# rows deleted JDM 8/90
000150      TSB#SRT DS  F   ..# sorts performed JDM 8/90
000154      TSB#SRR DS  F   ..# rows sorted JDM 8/90
000158      TSB#SMI DS  F   ..minimum rows sorted JDM 8/90
00015C      TSB#SMX DS  F   ..maximum rows sorted JDM 8/90
000160      TSB#AMC DS  F   ..# AM recompiles JDM 8/90
000164      DS   8F          ..reserved for expansion JDM 8/90
000184      TSBSQEND DS  0F   End of SQL statistics JDM 8/90
                  PRINT GEN
00184   TSBDLEN EQU  *-TSB LENGTH OF DSECT.
*****
***           ***
***   FOLLOWING THE TSB IS THE ASAB WHICH CONTAINS THE ADS ***
***   STATS FIELDS. THESE FIELDS IMMEDIATELY FOLLOW THE TSB. ***
***   SEE THE ASAB DSECT IN THE #ASBDS COPY MEMBER. ***
***           ***
*****           ***
000184   ASABHERE DS  0F          THE ASAB STARTS HERE

```

1.23 #ASQDS

```

COPY #ASQDS
*****
***          ADSA SESSION QUEUE HEADER RECORD      ***
*** -----
***  THIS IS ALWAYS THE FIRST RECORD IN AN ADSA GENERATOR SESSION  ***
***  QUEUE. THIS RECORD IS USED PRIMARILY TO PROVIDE POSITIVE    ***
***  IDENTIFICATION OF THE QUEUES.                                ***
*****




| <u>Offset</u>                                                          | <u>Value</u> |                  |                                      |
|------------------------------------------------------------------------|--------------|------------------|--------------------------------------|
| 000000                                                                 | AQH          | DSECT            | 12:20:32 04/04/85                    |
| 000000                                                                 | AQHID        | DS CL4           | 'AQH*'                               |
| 000004                                                                 | AQHANAME     | DS CL8           | APPLICATION NAME                     |
| 00000C                                                                 | AQHAVER      | DS H             | APPLICATION VERSION                  |
| 00000E                                                                 | AQHIMPL      | DS X             | APPLICATION IMPLICITLY CHECK OUT GWG |
| 00000F                                                                 |              | DS X             | RESERVED                             |
| 000010                                                                 | AQHNODE      | DS CL8           | DICTIONARY NODE NAME                 |
| 000018                                                                 | AQHDIRCT     | DS CL8           | DICTIONARY DBNAME                    |
| 000020                                                                 | AQHUSRID     | DS CL32          | USER ID (OWNER OF THIS SESSION)      |
| 000040                                                                 | AQHADBDT     | DS CL8           | ADB GENERATE DATE                    |
| 000048                                                                 | AQHADBTM     | DS CL8           | ADB GENERATE TIME                    |
| 000050                                                                 | AQHLEN       | EQU *-AQH        | SIZE OF SESSION QUEUE HEADER         |
| 000014                                                                 | AQHLENW      | EQU (AQHLEN+3)/4 | SIZE OF SESSION QUEUE HDR IN WORDS   |
| *****                                                                  |              |                  |                                      |
| ***          ADSA SESSION QUEUE RECORD      ***                        |              |                  |                                      |
| *** -----                                                              |              |                  |                                      |
| ***  THESE QUEUE RECORDS ARE OF VARIABLE LENGTH. ONE RECORD IS   ***   |              |                  |                                      |
| ***  WRITTEN FOR EACH SUCCESSFUL UPDATE TRANSACTION ENTERED BY THE *** |              |                  |                                      |
| ***  USER. EACH RECORD CONTAINS THE FUNCTION CODE WITH WHICH THE ***   |              |                  |                                      |
| ***  SCREEN INTERFACE WAS CALLED, AND ANY PARMS OTHER THAN     ***     |              |                  |                                      |
| ***  ADSOAWRK AND ADSOAUNI.                                ***         |              |                  |                                      |
| *****                                                                  |              |                  |                                      |
| 000000                                                                 | AQR          | DSECT            |                                      |
| 000000                                                                 | AQRID        | DS CL4           | 'AQR*'                               |
| 000004                                                                 | AQRPRMCT     | DS H             | NUMBER OF PARMS IN THIS CALL         |
| 000006                                                                 | AQRSFNC1     | DS H             | AGNP ROUTING CODE                    |
| * THE FOLLOWING FIELDS ARE MODULE SPECIFIC                             |              |                  |                                      |
| 000008                                                                 | AQRSFNC2     | DS H             | FUNCTION CODE                        |
| 00000A                                                                 | AQRRAFOPR    | DS X             | OPERATION CODE                       |
| 00000B                                                                 | AQRRAFOCC    | DS X             | RECORD OCCURRENCE                    |
| 00000C                                                                 | AQRGABSK     | DS F             | STORAGE KEY OF THE GAB               |
| 000010                                                                 | AQRCRNCY     | DS 10F           | CURRENCY VECTOR AT TIME OF CALL      |
| 000038                                                                 | AQRPARMS     | DS 0F            | START OF THE VARIABLE PORTION        |
| 000038                                                                 | AQRPFXLN     | EQU *-AQRID      | LENGTH OF AQR PREFIX                 |


```

1.24 #ASRDS

```
COPY #ASRDS
*****
***  ASR IS A DSECT DEFINING THE AUTOSTATUS RECORD CONTROL BLOCK ***
***  FOR THE FDB OF AN ADSO DIALOG.  THIS ONLY EXISTS WHEN AN      ***
***  AUTOSTATUS RECORD HAS BEEN NAMED FOR THE DIALOG.  IT IS        ***
***  ANCHORED IN FDBASRA.                                         ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-----------|
| 000000 | ASR | DSECT |
| 000000 | ASRNAME | DS CL32 |
| 000020 | ASRVER | DS H |
| 000024 | | DS 4F |
| 00034 | ASRLEN | EQU *-ASR |

21:21:53 03/31/82

NAME OF THE AUTOSTATUS RECORD
RECORD VERSION
RESERVED

LENGTH OF ASR FIXED PORTION

1.25 #ATCDS

```
COPY #ATCDS
*****
***      ATC:  ADS APPLICATION GENERATOR TASK CODE RECORD
***      ATC IS A DSECT THAT DESCRIBES THE FIELDS IN THE
***      ADSO-APPL-TASK-CODES RECORD USED BY THE ADS APPLICATION
***      GENERATOR.  THIS RECORD IS CURRENTLY STORED AS A
***      SYSCMT-038 RECORD (TYPE -17).
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--------|
| 000000 | ATC | DSECT |
| 000000 | ATCSEQ | DS F |
| 000004 | ATCNAME | DS CL8 |
| 00000C | ATCTOPF | DS CL8 |
| 000014 | | DS 84X |
| 000068 | ATCTYPE | DS F |

11:40:14 10/05/82

SYSCMT-038 SEQUENCE NUMBER

TASK CODE NAME

TOP FUNCTION FOR THIS TASK NAME

FILLER

TYPE OF SYSCMT-038 RECORD (-17)

1.26 #AWERRS

```
COPY #AWERRS
*****
***          EQUATES FOR AWERRS VALUES      ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|----------------------------------|
| 000F0 | AWNOERRS EQU X'F0' | NO ERRORS DETECTED |
| 000F1 | AWARNING EQU X'F1' | WARNING. DEFINITION CAN CONTINUE |
| 000F2 | AWERROR EQU X'F2' | ERROR. TERMINATE DEFINITION |
| 000F3 | AWFATAL EQU X'F3' | FATAL ERROR DETECTED |
| 000F4 | AWNONTFND EQU X'F4' | RECORD (CNTL-BLK) NOT FOUND |
| 000F5 | AWENDSET EQU X'F5' | END-OF-SET (CNTL-BLK CHAIN) |
| 000F6 | AWINVACT EQU X'F6' | INVALID ACTION |
| 000F7 | AWBADAPL EQU X'F7' | BAD APPLICATION |

1.27 #BDSDS

```

COPY #BDSDS
*****
***      BDS: BATCH DATABASE STATISTICS BLOCK
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------------|------------------------------------|
| 000000 | BDS DSECT | 09:33:41 04/15/86 |
| 000000 | BDSBID DS CL4 | 'BDS*' |
| 000004 | BDSID DS 0CL40 | BATCH DATABASE STATS ID |
| 000004 | BDSUSID DS CL32 | USER ID |
| 000024 | BDSBATID DS CL8 | "*BATCH**" |
| 00002C | BDSUID DS CL8 | DIALOG NAME OR "\$ADS@AO" |
| 000034 | BDSUDAT DS PL4 | DATE STATISTICS STARTED |
| 000038 | BDSUTIM DS F | TIME STATISTICS STARTED |
| 00003C | BDSDB DS OF | START OF DB STATISTICS. |
| 00003C | BDSPAGRD DS F | NUMBER OF PAGES READ. |
| 000040 | BDSPAGWR DS F | NUMBER OF PAGES WRITTEN. |
| 000044 | BDSPAGRQ DS F | NUMBER OF PAGES REQUESTED. |
| 000048 | BDSCALNO DS F | NUMBER OF CALC RECS WITH NO OFLOW. |
| 00004C | BDSCALOF DS F | NUMBER OF CALC RECS WITH OFLOW. |
| 000050 | BDSVIANO DS F | NUMBER OF VIA RECS WITH NO OFLOW. |
| 000054 | BDSVIAOF DS F | NUMBER OF VIA RECS WITH OFLOW. |
| 000058 | BDSRECRQ DS F | NUMBER OF RECORDS REQUESTED. |
| 00005C | BDSRECCU DS F | NUMBER OF RECS CURRENT OF R-U. |
| 000060 | DS F | NOT USED. |
| 000064 | BDSFRAGS DS F | NUMBER OF FRAGMENTS STORED. |
| 000068 | BDSRELO DS F | NUMBER OF RECORDS RELOCATED. |
| 00006C | BDSTLOCK DS F | TOTAL NUMBER OF LOCKS |
| 000070 | BDSSLOCK DS F | NUMBER OF SELECT LOCKS |
| 000074 | BDSULOCK DS F | NUMBER OF UPDATE LOCKS |
| 000078 | 0003C BDOSCG EQU *-BDSDB | OFFSET TO GET SCRATCH |
| 000078 | BDSESCG DS F | NUMBER SCRATCH GETS |
| 00007C | 00040 BDOSCP EQU *-BDSDB | OFFSET TO PUT SCRATCH |
| 00007C | BDSESCP DS F | NUMBER SCRATCH PUTS |
| 000080 | 00044 BDOSCD EQU *-BDSDB | OFFSET TO DEL SCRATCH |
| 000080 | BDSESCD DS F | NUMBER SCRATCH DELETES |
| 000084 | 00048 BDSEQUG EQU *-BDSDB | OFFSET TO GET QUEUE |
| 000084 | BDSEQUG DS F | NUMBER QUEUE GETS |
| 000088 | 0004C BDSEQUP EQU *-BDSDB | OFFSET TO PUT QUEUE |
| 000088 | BDSEQUP DS F | NUMBER QUEUE PUTS |
| 00008C | 00050 BDSEQUD EQU *-BDSDB | OFFSET TO DEL QUEUE |
| 00008C | BDSEQUUD DS F | NUMBER QUEUE DELETES |
| 000094 | BDSDSLEN EQU *-BDSDB | LENGTH OF DATA BLOCK |
| 000090 | BDSDSECL EQU *-BDS | LENGTH OF DSECT |

1.28 #BLEDS

```

COPY #BLEDS
*****
*          ---BATCH FILE LIST ELEMENT DSECT---
*
* THIS FILE MAPPING CONTROL BLOCK IS MANIPULATED BY THE DIRECT-CALL
* PORTION OF THE FILE MAPPING RUNTIME SYSTEM. THE BLE CHAIN IS
* ANCHORED IN THE BMVT AT BMVTBLEA.
*
* THE BLE CONTAINS INFORMATION ABOUT A DATASET AT THE OPERATING
* SYSTEM LEVEL AND IS OF VARIABLE LENGTH DEPENDING UPON
* THE BLE DSORG (TYPE).
*
* BLE TYPES ARE:
*
*   1) SYSIN      (OS|DOS)
*   2) SYSOUT     (OS|DOS)
*   3) CARD       (DOS)
*   4) PRINT      (DOS)
*   5) PS         (OS|DOS|BS2K|CMS)
*   6) VSAM      (OS|DOS|CMS)
*
* A BLE IS CREATED WHEN A DATASET IS PHYSICALLY OPENED, AND FREED
* WHEN A DATASET IS PHYSICALLY CLOSED.
*
*****

```

| <u>Offset</u> | <u>Value</u> | | | |
|---------------|--------------|-------------|--|------------------------|
| 000000 | BLE | DSECT | 11:24:42 02/04/87 09/27/90 | |
| 000000 | BLEID | DS CL4 | EYECATCHER 'BLE*' | |
| 000004 | BLENEXT | DS A | A(NEXT BLE) | |
| 000008 | BLELEN | DS H | LENGTH OF THIS BLE | |
| 00000A | BLEUCNT | DS H | BLE USE COUNT (#BRPLS USING BLE) | |
| | * | | | |
| | BLEPS | #FLAG X'80' | FILE TYPE IS PHYSICAL SEQUENTIAL | |
| 00000C | BLEPSI | DS 0XL1 | | |
| | 000080 | BLEPSM | EQU X'80' | |
| | | BLESI | #FLAG X'40' | FILE TYPE IS SYSIN |
| 00000C | | BLESII | DS 0XL1 | |
| | 000040 | BLESIM | EQU X'40' | |
| | | BLECD | #FLAG X'20' | FILE TYPE IS CARD |
| 00000C | | BLECDI | DS 0XL1 | |
| | 000020 | BLECDM | EQU X'20' | |
| | | BLESO | #FLAG X'10' | FILE TYPE IS SYSOUT |
| 00000C | | BLESOI | DS 0XL1 | |
| | 000010 | BLESOM | EQU X'10' | |
| | | BLEPR | #FLAG X'08' | FILE TYPE IS PRINT |
| 00000C | | BLEPRI | DS 0XL1 | |
| | 000008 | BLEPRM | EQU X'08' | |
| | | BLEKS | #FLAG X'04' | RESERVED FOR VSAM KSDS |
| 00000C | | BLEKSI | DS 0XL1 | |
| | 000004 | BLEKSM | EQU X'04' | |
| | | BLEES | #FLAG X'02' | FILE TYPE IS VSAM ESSD |
| 00000C | | BLEESI | DS 0XL1 | |
| | 000002 | BLEESM | EQU X'02' | |
| | | BLERR | #FLAG X'01' | RESERVED FOR VSAM RRDS |
| 00000C | | BLERRI | DS 0XL1 | |
| | 000001 | BLERRM | EQU X'01' | |
| | | BLEVS | #FLAG BLEKSM+BLEESM+BLERRM FILE TYPE IS VSAM | |
| 00000C | | BLEVSI | DS 0XL1 | |
| | 000007 | BLEVSM | EQU BLEKSM+BLEESM+BLERRM | |
| 00000C | | BLEDSO | DS X | BLE DSORG (FILE TYPE) |

| | | | |
|---|----------|-------------|---|
| 00000D | DS | X | RESERVED |
| 00000E | BLEPFLG1 | DS | X PREFIX FLAG BYTE 1 |
| * | | | |
| | BLEOS | #FLAG X'80' | OPERATING SYSTEM IS OS |
| 00000F | BLEOSI | DS | 0XL1 |
| | 00080 | BLEOSM | EQU X'80' |
| | | BLEOS | #FLAG X'40' |
| 00000F | BLEDESI | DS | 0XL1 |
| | 00040 | BLEDESM | EQU X'40' |
| | | BLEBS2K | #FLAG X'20' |
| 00000F | BLEBS2KI | DS | 0XL1 |
| | 00020 | BLEBS2KM | EQU X'20' |
| | | BLECMS | #FLAG X'10' |
| 00000F | BLECMSSI | DS | 0XL1 |
| | 00010 | BLECMMSM | EQU X'10' |
| 00000F | BLEOPSYS | DS | X OPERATING SYSTEM INDICATORS |
| 000010 | | DS | 2H RESERVED |
| | 00014 | BLEPFXL | EQU ((*-BLE)/4)*4 LENGTH OF BLE PREFIX |
| <hr/> * THE FUNCTION OF THE BLE HEADER IS TO MAINTAIN INFORMATION COMMON * TO ALL ACCESS METHODS AND OPERATING SYSTEMS. <hr/> | | | |
| 000014 | BLEHDR | DS | OF BEGIN BLE HEADER |
| | BLEEOF | #FLAG X'80' | END-OF-FILE REACHED FOR DATASET |
| 000014 | BLEEOFI | DS | 0XL1 |
| | 00080 | BLEEOFM | EQU X'80' |
| | | BLESYND | #FLAG X'40' |
| 000014 | BLESYNDI | DS | 0XL1 SYNAD EXIT WAS TAKEN FOR DATASET |
| | 00040 | BLESYNDM | EQU X'40' |
| | | BLETEMP | #FLAG X'20' |
| 000014 | BLETEMPI | DS | 0XL1 TEMPORARY DATASET |
| | 00020 | BLETEMPM | EQU X'20' |
| | | BLEPDS | #FLAG X'10' |
| 000014 | BLEPDSI | DS | 0XL1 MEMBER OF PARTITIONED DATASET |
| | 00010 | BLEPDSM | EQU X'10' |
| | | BLEDMY | #FLAG X'08' |
| 000014 | BLEDMYI | DS | 0XL1 DUMMY (OS) IGN (DOS) |
| | 00008 | BLEDMYM | EQU X'08' |
| | | BLEGU | #FLAG X'04' |
| 000014 | BLEGUI | DS | 0XL1 GET-UPDATE PERFORMED (USAGE I/O) |
| | 00004 | BLEGUM | EQU X'04' |
| | | BLEUPD | #FLAG X'02' |
| 000014 | BLEUPDI | DS | 0XL1 VSAM RPL HAS OPTCD=UPD |
| | 00002 | BLEUPDM | EQU X'02' |
| | | BLELER | #FLAG X'01' |
| 000014 | BLELERI | DS | 0XL1 LERAD EXIT WAS TAKEN FOR DATASET |
| | 00001 | BLELERM | EQU X'01' |
| 000014 | BLEFSTAT | DS | X FILE STATUS INDICATOR |
| | | BLEOPEN | #FLAG X'01' |
| 000015 | BLEOPENI | DS | 0XL1 OPEN REQUEST |
| | 00001 | BLEOPENM | EQU X'01' |
| | | BLECLOS | #FLAG X'02' |
| 000015 | BLECLOSI | DS | 0XL1 CLOSE REQUEST |
| | 00002 | BLECLOSM | EQU X'02' |
| | | BLEREAD | #FLAG X'03' |
| 000015 | BLEREADI | DS | 0XL1 READ REQUEST |
| | 00003 | BLEREADM | EQU X'03' |
| | | BLEWRIT | #FLAG X'04' |
| 000015 | BLEWRITI | DS | 0XL1 WRITE REQUEST |
| | 00004 | BLEWRITM | EQU X'04' |
| | | BLEPOIN | #FLAG X'05' |
| 000015 | BLEPOINI | DS | 0XL1 POINT REQUEST |
| | 00005 | BLEPOINM | EQU X'05' |
| | | BLEBUF | #FLAG X'06' |
| 000015 | BLEBUFI | DS | 0XL1 READ WRITE BUFFER REQ (DSORG=PS)) |
| | 00006 | BLEBUFM | EQU X'06' |

| | | | |
|--------|----------|------------------------------------|----------------------------------|
| | BLEREC | #FLAG BLEBUF M | READ WRITE RECORD REQUEST (VSAM) |
| 000015 | BLERECI | DS 0XL1 | |
| | BLERECM | EQU BLEBUF M | |
| | BLEEXDS | #FLAG X'07' | EXAMINE DATASET |
| 000015 | BLEEXDSI | DS 0XL1 | |
| | BLEEXDSM | EQU X'07' | |
| | * | | |
| 000015 | BLEOPER | DS X | OPERATION REQUEST BYTE |
| | BLEOINP | #FLAG X'80' | FILE WAS OPENED FOR INPUT |
| 000016 | BLEOINPI | DS 0XL1 | |
| | BLEOINPM | EQU X'80' | |
| | BLEOOUT | #FLAG X'40' | FILE WAS OPENED FOR OUTPUT |
| 000016 | BLEOOUTI | DS 0XL1 | |
| | BLEOOUTM | EQU X'40' | |
| | BLEOIO | #FLAG BLEOINPM+BLEOOUTM | FILE IS OPENED 'I/O' |
| 000016 | BLEOIOI | DS 0XL1 | |
| | BLEOIAM | EQU BLEOINPM+BLEOOUTM | |
| 000016 | BLEUSAGE | DS X | DATASET USAGE |
| | * | | |
| | BLEBLK | #FLAG X'10' | BLOCKED RECORDS |
| 000017 | BLEBLKI | DS 0XL1 | |
| | BLEBLKM | EQU X'10' | |
| | BLESB | #FLAG X'08' | STANDARD BLOCKS(RECFM F ONLY) |
| 000017 | BLESBI | DS 0XL1 | |
| | BLESBM | EQU X'08' | |
| | BLEASA | #FLAG X'04' | ANSI (ASA) CARRIAGE CONTROL |
| 000017 | BLEASAI | DS 0XL1 | |
| | BLEASAM | EQU X'04' | |
| | BLEMACH | #FLAG X'02' | MACHINE CODE CARRIAGE CONTROL |
| 000017 | BLEMACHI | DS 0XL1 | |
| | BLEMACHM | EQU X'02' | |
| | BLECC | #FLAG BLEASAM+BLEMACHM | CARRIAGE CONTROL PRESENT |
| 000017 | BLECCI | DS 0XL1 | |
| | BLECCM | EQU BLEASAM+BLEMACHM | |
| | * | | |
| | BLEF | #FLAG X'80' | FIXED-UNBLOCKED |
| 000017 | BLEFI | DS 0XL1 | |
| | BLEFM | EQU X'80' | |
| | BLEFS | #FLAG BLEFM+BLESBM | FIXED-UNBLOCKED STD |
| 000017 | BLEFSI | DS 0XL1 | |
| | BLEFSM | EQU BLEFM+BLESBM | |
| | BLEFA | #FLAG BLEFM+BLEASAM | FIXED-UNBLOCKED ASA |
| 000017 | BLEFAI | DS 0XL1 | |
| | BLEFAM | EQU BLEFM+BLEASAM | |
| | BLEFSA | #FLAG BLEFM+BLEASAM+BLESBM | FIXED-UNBLOCKED ASA STD |
| 000017 | BLEFSAI | DS 0XL1 | |
| | BLEFSAM | EQU BLEFM+BLEASAM+BLESBM | |
| | BLEFM | #FLAG BLEFM+BLEMACHM | FIXED-UNBLOCKED MACH |
| 000017 | BLEFMI | DS 0XL1 | |
| | BLEFMM | EQU BLEFM+BLEMACHM | |
| | BLEFSM | #FLAG BLEFM+BLEMACHM+BLESBM | FIXED-UNBLOCKED MACH STD |
| 000017 | BLEFSMI | DS 0XL1 | |
| | BLEFSMM | EQU BLEFM+BLEMACHM+BLESBM | |
| | BLEFB | #FLAG BLEFM+BLEBLKM | FIXED-BLOCKED |
| 000017 | BLEFBFI | DS 0XL1 | |
| | BLEFBM | EQU BLEFM+BLEBLKM | |
| | BLEFBFS | #FLAG BLEFM+BLEBLKM+BLESBM | FIXED-BLOCKED STD |
| 000017 | BLEFBSI | DS 0XL1 | |
| | BLEFBSM | EQU BLEFM+BLEBLKM+BLESBM | |
| | BLEFBBA | #FLAG BLEFM+BLEBLKM+BLEASAM | FIXED-BLOCKED ASA |
| 000017 | BLEFBAI | DS 0XL1 | |
| | BLEFBAM | EQU BLEFM+BLEBLKM+BLEASAM | |
| | BLEFBSA | #FLAG BLEFM+BLEBLKM+BLEASAM+BLESBM | FIXED-BLOCKED ASA STD |
| 000017 | BLEFBSAI | DS 0XL1 | |
| | BLEFBSAM | EQU BLEFM+BLEBLKM+BLEASAM+BLESBM | |

| | | | | |
|--------|----------|-----------|------------------------|--|
| | BLEFBM | #FLAG | BLEFM+BLEBLKM+BLEMACHM | FIXED-BLOCKED MACH |
| 000017 | BLEFBMI | DS | 0XL1 | |
| | 00092 | BLEFBMM | EQU | BLEFM+BLEBLKM+BLEMACHM |
| | | BLEFBSM | #FLAG | BLEFM+BLEBLKM+BLEMACHM+BLESBM FIXED-BLOCKED MACH STD |
| 000017 | | BLEFBSMI | DS | 0XL1 |
| | 0009A | BLEFBSSMM | EQU | BLEFM+BLEBLKM+BLEMACHM+BLESBM |
| | | * | | |
| | | BLEV | #FLAG | X'40' |
| 000017 | | BLEVI | DS | 0XL1 |
| | 00040 | BLEVM | EQU | X'40' |
| | | BLEVA | #FLAG | BLEVM+BLEASAM |
| 000017 | | BLEVAI | DS | 0XL1 |
| | 00044 | BLEVAM | EQU | BLEVM+BLEASAM |
| | | BLEVM | #FLAG | BLEVM+BLEMACHM |
| 000017 | | BLEVMI | DS | 0XL1 |
| | 00042 | BLEVMM | EQU | BLEVM+BLEMACHM |
| | | BLEVB | #FLAG | BLEVM+BLEBLKM |
| 000017 | | BLEVBI | DS | 0XL1 |
| | 00050 | BLEVBM | EQU | BLEVM+BLEBLKM |
| | | BLEVBA | #FLAG | BLEVM+BLEBLKM+BLEASAM |
| 000017 | | BLEVBAI | DS | 0XL1 |
| | 00054 | BLEVBAM | EQU | BLEVM+BLEBLKM+BLEASAM |
| | | BLEVBM | #FLAG | BLEVM+BLEBLKM+BLEMACHM |
| 000017 | | BLEVBMI | DS | 0XL1 |
| | 00052 | BLEVBMM | EQU | BLEVM+BLEBLKM+BLEMACHM |
| | | * | | |
| | | BLEU | #FLAG | BLEFM+BLEVM |
| 000017 | | BLEUI | DS | 0XL1 |
| | 000C0 | BLEUM | EQU | BLEFM+BLEVM |
| | | BLEUA | #FLAG | BLEFM+BLEVM+BLEASAM |
| 000017 | | BLEUAI | DS | 0XL1 |
| | 000C4 | BLEUAM | EQU | BLEFM+BLEVM+BLEASAM |
| | | BLEUM | #FLAG | BLEFM+BLEVM+BLEMACHM |
| 000017 | | BLEUMI | DS | 0XL1 |
| | 000C2 | BLEUMM | EQU | BLEFM+BLEVM+BLEMACHM |
| | | * | | |
| 000017 | | BLERECFM | DS | X |
| | | * | | RECORD FORMAT |
| | | | | |
| 000018 | BLEFILEL | DS | X | LENGTH OF FILE ID |
| 000019 | BLEFILE | DS | OCL8 | FILE ID |
| 000019 | BLEDDNAM | DS | CL8 | DDNAME (OS) |
| 000021 | BLESYS | DS | X | LENGTH OF SYSTEM NUMBER |
| 000022 | BLESYS | DS | CL6 | LOGICAL SYSTEM NUMBER (DOS/VSE) |
| 000028 | BLEVOLSL | DS | X | LENGTH OF VOL SER |
| 000029 | BLEVOLS | DS | CL6 | VSN OF 1ST DATASET VOLUME (OS) |
| 00002F | BLEMSG# | DS | PL4 | LEVEL 3 MSG ID FROM I/O MODULE |
| 000033 | BLEDSNML | DS | X | LENGTH OF DATASET NAME (#WTL) |
| 000034 | BLEDSNM | DS | CL54 | DATASET NAME |
| 00006A | ORG | BLEDSNM | | |
| 000034 | BLEBDSNM | DS | CL54 | BS2000 DATASET NAME |
| 00006A | ORG | BLEDSNM | | |
| 000034 | BLEODSNM | DS | CL44 | OS DATASET NAME |
| 000060 | DS | C | | LEFT PAREN |
| 000061 | BLEMBRNM | DS | CL8 | MEMBER NAME IF PARTITIONED DATASET |
| 000069 | DS | C | | RIGHT PAREN |
| 00006A | ORG | BLEDSNM | | |
| 000034 | BLEDDSNM | DS | CL44 | DOS DATASET NAME |
| 000060 | DS | CL10 | | |
| 00006A | DS | H | | RESERVED BY FILE MAPPING |
| 00006C | BLEDEVNM | DS | CL4 | DEVICE NAME (DOS) |
| | | * | | |
| 000070 | BLEIORBA | DS | A | A(DEFRB) FOR DOS/VSE |
| 000074 | ORG | BLEIORBA | | |
| 000070 | BLEACBA | DS | A | A(ACB) FOR OS DOS VSAM |
| 000074 | ORG | BLEIORBA | | |
| 000070 | BLEDCBA | DS | A | A(DCB) (OS) |

| | | | |
|--------|----------|----------------------|------------------------------------|
| | | * | |
| 000074 | BLEDECBA | DS A | A(DATA EVENT CONTROL BLOCK) (OS) |
| 000078 | | ORG BLEDECBA | |
| 000074 | BLERPLA | DS A | A(RPL) FOR OS DOS VSAM |
| | | * | |
| 000078 | BLEDFR15 | DS F | R15 R/C FROM DOS FILE I/O |
| 00007C | BLER15 | DS F | R15 R/C FROM I/O MODULES |
| 000080 | BLESYNMS | DS A | A(SYNAD ERROR ANALYSIS MESSAGE) |
| 000084 | BLEIOBFA | DS A | A(I/O BUFFER STORAGE) |
| 000088 | BLEC1OBA | DS A | A(CURRENT I/O BUFFER) |
| 00008C | BLE#INP | DS F | # PHYSICAL INPUT OPERATIONS |
| 000090 | BLE#OUT | DS F | # PHYSICAL OUTPUT OPERATIONS |
| 000094 | BLEMXPRL | DS 0F | MAXIMUM PHYSICAL RECORD LENGTH |
| 000094 | BLEBUFL | DS 0F | BUFFER ELEMENT SIZE (PHYS REC LEN) |
| 000094 | BLERECLM | DS F | VSAM MAXIMUM LRECL (FROM OPEN) |
| 000098 | | ORG BLEBUFL | |
| 000094 | | DS H | |
| 000096 | BLEOBLKS | DS H | NON-VSAM BLOCK SIZE (FROM OPEN) |
| 000098 | BLEMLRL | DS 0F | VSAM RECORD SIZE (FROM OPEN) |
| 000098 | | DS H | |
| 00009A | BLEORECL | DS H | NON-VSAM RECORD SIZE (FROM OPEN) |
| 00009C | BLERECLR | DS 0F | VSAM RECORD SIZE (READ WRITE) |
| 00009C | | DS H | |
| 00009E | BLELRECL | DS H | NON-VSAM RECORD SIZE (READ WRITE) |
| 0000A0 | BLERECLV | DS 0F | VSAM LAST MODCB RECORD LENGTH |
| 0000A0 | | DS H | |
| 0000A2 | BLEBLKSI | DS H | NON-VSAM BLOCK SIZE (READ WRITE) |
| 0000A4 | BLEBRPLA | DS A | A(LAST BRPL SERVICED BY THIS BLE) |
| 0000A8 | BLERECA | DS A | A(LAST LOGICAL REC READ/WRITTEN) |
| 0000AC | BLELISTA | DS A | A(GENERAL USE PARMLIST) |
| 0000B0 | BLEIOSL | DS F | I/O MODULE ACQUIRED STG LENGTH |
| 0000B4 | BLEFMDEA | DS F | A(FMD TABLE ENTRY) FOR THIS DSORG |
| 0000B8 | BLEDEVT | DS X | OS DEVICE TYPE |
| | BLEBPO | #FLAG X'80' | BCTL-BYPASS PARALLEL OPEN CHECK |
| 0000B9 | BLEBPOI | DS 0XL1 | |
| | 00080 | BLEBPOM EQU X'80' | |
| | | BLECCR #FLAG X'40' | RECFM ERROR IN CONCATENATION |
| 0000B9 | BLECCRI | DS 0XL1 | |
| | 00040 | BLECCRM EQU X'40' | |
| | | BLECCB #FLAG X'20' | BLKSIZE ERROR IN CONCATENATION |
| 0000B9 | BLECCBI | DS 0XL1 | |
| | 00020 | BLECCBM EQU X'20' | |
| | | BLECCL #FLAG X'10' | LRECL ERROR IN CONCATENATION |
| 0000B9 | BLECCLI | DS 0XL1 | |
| | 00010 | BLECCLM EQU X'10' | |
| | | BLERRD #FLAG X'08' | REISSUE OS READ MACRO |
| 0000B9 | BLERRDI | DS 0XL1 | |
| | 00008 | BLERRDM EQU X'08' | |
| | | BLESYNC #FLAG X'04' | SYNCHRONIZE ALL PHYSICAL I/O |
| 0000B9 | BLESYNCI | DS 0XL1 | |
| | 00004 | BLESYNCM EQU X'04' | |
| | | BLEPEOF #FLAG X'02' | INFORM SUBTASK EOF REACHED |
| 0000B9 | BLEPEOFI | DS 0XL1 | |
| | 00002 | BLEPEOFM EQU X'02' | |
| | | BLEPSYN #FLAG X'01' | INFORM SUBTASK OF PHYS I/O ERROR |
| 0000B9 | BLEPSYNI | DS 0XL1 | |
| | 00001 | BLEPSYNM EQU X'01' | |
| 0000B9 | BLEHFLG1 | DS X | HEADER FLAG BYTE |
| 0000BA | BLEBUFNO | DS H | NUMBER OF BUFFER ELEMENTS |
| 0000BC | BLEBIOCT | DS H | # BUFFERS SCHEDULED FOR I/O |
| 0000BE | BLEAMRBL | DS X | LENGTH OF A REQUEST BLOCK |
| 0000BF | | DS X | RESERVED BY FILE MAPPING |
| 0000C0 | | DS 4F | RESERVED BY FILE MAPPING |
| 000BC | BLEHDRLN | EQU ((*-BLEHDR)/4)*4 | LENGTH OF BLE HEADER AREA |
| 0000D0 | BLEVAR | DS 0F | BEGIN BLE VARIABLE SECTION |
| | | | ***** |

```

* PHYSICAL SEQUENTIAL DATASET SECTION
*
* SYSIN DATASET SECTION
*
* SYSOUT DATASET SECTION
*
* CARD DATASET SECTION
*
* PRINT DATASET SECTION
*
* SAM DATASET SECTION (BS2000)
*
* ISAM DATASET SECTION (BS2000)
*
*****
0000D0    BLENPOS  DS   F          DISPLACEMENT INTO BLOCK
0000D4    BLECUU   DS   CL3       ACTUAL DEVICE ADDRESS (CUU)(DOS)
0000D7    BLECC#   DS   X          CONCATENATION NUMBER (OS)
0000D8    BLEORCFM DS   X         OVERRIDE RECFM
0000D9    BLEOUNIT DS   X         OVERRIDE UNIT
*
* BLEADASD #FLAG X'80'           DEVICE IS DIRECT-ACCESS
0000DA    BLEADASDI DS  OXL1
0000DA    00080  BLEADASDM EQU X'80'
* BLEMT #FLAG X'40'             DEVICE IS A MAGNETIC TAPE
0000DA    BLEMTI   DS  OXL1
0000DA    00040  BLEMTM   EQU X'40'
0000DA    BLEUNIT  DS   X          UNIT (DEVICE CLASS)
0000DB    DS   X          RESERVED
0000DC    DS   H          RESERVED
0000DE    BLENBLKS DS   H         # PHYSICAL I/O BLOCKS IN DOS/VS(E)
0000E0    DS   2F        RESERVED
0000E8    BLESOLEN EQU (((*-BLE)/4)*4) LENGTH OF SYSOUT ELEMENT
0000E8    BLECDLEN EQU (((*-BLE)/4)*4) LENGTH OF CARD ELEMENT
0000E8    BLEPRLEN EQU (((*-BLE)/4)*4) LENGTH OF PRINT ELEMENT
0000E8    BLESILEN EQU (((*-BLE)/4)*4) LENGTH OF SYSIN ELEMENT
0000E8    BLEULPNM DS  CL8       NAME OF USER LABEL PROGRAM (TAPE)
0000F0    ORG  BLEULPNM
0000E8    BLEUHLPA DS  A          A(TAPE USER HEADER LABEL PGM)
0000EC    BLEUTLPA DS  A          A(TAPE USER TRAILER LABEL PGM)
0000F0    BLETRR   DS  F          TTR OF BLOCK JUST READ OR WRITTEN
*
* #FLAG X'80'                  RESERVED (IBM)
* BLEAL #FLAG X'40'            ANSI LABEL (OS)
0000F4    BLEALI   DS  OXL1
0000F4    00040  BLEALM   EQU X'40'
* BLEAUL #FLAG X'44'            ANSI LABEL + USER LABEL (OS)
0000F4    BLEAULI  DS  OXL1
0000F4    00044  BLEAULM  EQU X'44'
* BLEBLP #FLAG X'10'            BYPASS LABEL PROCESSING (OS)
0000F4    BLEBLPI DS  OXL1
0000F4    00010  BLEBLPM  EQU X'10'
* BLESUL #FLAG X'0A'            STANDARD LABEL + USER LABEL
0000F4    BLESULI  DS  OXL1
0000A    BLESULM EQU X'0A'
* BLENSL #FLAG X'04'            NON-STANDARD LABEL
0000F4    BLENSLI  DS  OXL1
0000F4    00004  BLENSLM  EQU X'04'
* BLESL #FLAG X'02'              STANDARD LABEL
0000F4    BLESLI   DS  OXL1
0000F4    00002  BLESLM   EQU X'02'
* BLENL #FLAG X'01'              NO LABEL
0000F4    BLENLI   DS  OXL1
0000F4    00001  BLENLM   EQU X'01'
* BLEFLABL DS   X              TAPE FILE LABEL TYPE
0000F5    DS   X              RESERVED

```

| | | | |
|------------------------------|---------------------|---------------|------------------------------------|
| 0000F6 | DS | X | RESERVED |
| 0000F7 | DS | X | RESERVED |
| * | | | |
| 0000F8 | BLEBS2K1 DS | X | B2SK: FLAG BYTE 1 |
| 000080 | BLE1SAM EQU | X'80' | FILE TYPE = SAM |
| 000040 | BLE1INSTD EQU | X'40' | NON-STANDARD BLOCK SIZE |
| 000020 | BLE1DUPK EQU | X'20' | DUPLICATE KEY ALLOWED |
| * | | | |
| 0000F9 | BLEBS2K2 DS | X | BS2K: FLAG BYTE 2 |
| 000080 | BLE2RCF1 EQU | X'80' | RECFORM SPECIFIED IN /FILE |
| 000040 | BLE2RCF2 EQU | X'40' | WCC SPECIFIED IN /FILE |
| 000020 | BLE2BLKS EQU | X'20' | BLKSIZE SPECIFIED IN /FILE |
| 000010 | BLE2RECS EQU | X'10' | RECSIZE SPECIFIED IN /FILE |
| * | | | |
| 0000FA | BLEOPENT DS | X | B2SK: OPEN TYPE (EQU=ID1OPEN) |
| 0000FB | BLEBSRCF DS | X | BS2K: RECFORM FROM TFT OR CATALOG |
| 0000FC | BLEBSWCC DS | X | BS2K: WCC FROM TFT OR CATALOG |
| 0000FD | BLEKEYLE DS | X | BS2K: KEYLEN VALUE |
| 0000FE | BLEBSBLK DS | H | BS2K: BLKSIZE FROM TFT OR CATALOG |
| 000100 | BLEBSREC DS | H | BS2K: RECSIZE FROM TFT OR CATALOG |
| 000102 | BLEKEYPO DS | H | BS2K: KEYPOS VALUE |
| 000104 | BLEBSFRE DS | H | BS2K: REMAINING BUFR SPACE (SAM-V) |
| * | | | |
| 000108 | DS | 4F | RESERVED BY FILE MAPPING |
| 00118 | BLEPSLEN EQU | ((*-BLE)/4)*4 | LENGTH OF PHYS SEQUENTIAL ELEMENT |
| ***** | | | |
| * ---VSAM DATASET SECTION--- | | | |
| ***** | | | |
| 000118 | ORG | BLEVAR | |
| | BLEKSDS | #FLAG X'80' | CLUSTER IS KEY-SEQUENCED DATASET |
| 0000D0 | BLEKSDSI DS | 0XL1 | |
| | BLEKSDSM EQU | X'80' | |
| | BLEESDS #FLAG X'40' | | CLUSTER IS ENTRY-SEQUENCED DATASET |
| 0000D0 | BLEESDSI DS | 0XL1 | |
| | BLEESDSM EQU | X'40' | |
| | BLERRDS #FLAG X'20' | | CLUSTER IS RELATIVE-RECORD DATASET |
| 0000D0 | BLERRDSI DS | 0XL1 | |
| | BLERRDSM EQU | X'20' | |
| 0000D0 | BLEVSTYP DS | X | VSAM TYPE FLAGS |
| 0000D1 | DS | X | RESERVED |
| 0000D2 | DS | X | RESERVED |
| 0000D3 | DS | X | RESERVED |
| 0000D4 | BLEVKYEA DS | A | A(KEY) IF KSDS REL 10.2 |
| 0000D8 | ORG | BLEVKYEA | |
| 0000D4 | BLERRN DS | CL4 | REL REC NUM (RRDS) REL 10.2 |
| 0000D8 | BLEVKYO DS | F | OFFSET TO KEY REL 10.2 |
| 0000DC | BLEVKYL DS | F | KEY LENGTH REL 10.2 |
| 0000E0 | BLERBA DS | F | RBA OF CURRENT RECORD |
| 0000E4 | BLEFDBK DS | 0F | VSAM RPL FEEDBACK FIELD |
| 0000E4 | DS | X | RESERVED BY VSAM |
| 0000E5 | BLEFD15 DS | X | RETURN CODE FROM R15 |
| 0000E6 | BLEDFC DS | X | FUNCTION CODE |
| 0000E7 | BLEFDOD DS | X | FEEDBACK CODE |
| 0000E8 | BLEERROR DS | F | VSAM ACB ERROR FIELD |
| 0000EC | BLECBMRC DS | F | VSAM CONTROL BLOCK MACRO R/C (R0) |
| 0000F0 | BLERECAV DS | A | LAST MODCB AREA REQUEST |
| 0000F4 | BLEVSR15 DS | F | VSAM MACRO R15 RETURN CODE |
| 0000F8 | DS | 4F | RESERVED BY FILE MAPPING |
| 00108 | BLEVSLEN EQU | ((*-BLE)/4)*4 | LENGTH OF VSAM ELEMENT |

1.29 #BLPDS

```
COPY #BLPDS
*****
*** BLP:   BATCH LOG PREFIX DSECT
*** ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|------------------------------------|
| 000000 | BLP DSECT | 10:41:34 12/17/85 |
| 000000 | BLPDTK DS OCL8 | DATE/TIME (KEY) |
| 000000 | BLPDATE DS CL4 | DATE (00YYDDDC) |
| 000004 | BLPTIME DS CL4 | TIME (1/10000 SEC) |
| 000008 | BLPPGNAM DS CL8 | PROGRAM/DIALOG NAME |
| 000010 | BLPTYPE DS X | RECORD TYPE |
| | * THE FOLLOWING ARE VALID VALUES FOR BLPTYPE | |
| | * | |
| 00001 | BLPDBUG EQU 1 | DEBUG |
| 00002 | BLPUSR EQU 2 | USER MESSAGE |
| 00003 | BLPSPNSE EQU 3 | EDIT ERROR RECORD |
| 00004 | BLPWTO EQU 4 | WRITE TO OPERATOR MESSAGE |
| 00005 | BLPABND EQU 5 | ABEND DUMP TEXT |
| 00006 | BLPSTAT EQU 6 | STATISTICS |
| 00006 | BLPHITYP EQU 6 | HIGHEST VALID TYPE |
| | * | |
| | BLPRCMP #FLAG X'80' | IF ON, LOG RECORD IS COMPRESSED |
| 000011 | BLPRCMPI DS OXL1 | |
| 000080 | BLPRCMPM EQU X'80' | |
| 000011 | BLPFLG1 DS X | FLAG BYTE |
| 000012 | BLPRLEN DS H | LEN OF LOG REC THAT FOLLOWS PREFIX |
| 000014 | BLPULEN DS X | LENGTH OF USER ID |
| 000015 | DS XL3 | RESERVED |
| 000018 | BLPLEN EQU *-BLP | LENGTH OF FIXED PORTION OF PREFIX |
| 000018 | BLPUSER DS 0X | START OF USER ID |

1.30 #BMRBHDS

```
COPY #BMRBHDS
*****
***          BMRBH: BATCH MRB HEADER BLOCK
***
***      IN BATCH ADS, MRBS ARE ALLOCATED OUTSIDE OF THE VDB AND ***
***      AN MRB IS NOT FREED UNTIL THE FILE ASSOCIATED WITH IT IS ***
***      CLOSED. THE BMRBH DSECT DESCRIBES A HEADER WHICH PRECEDES ***
***      EACH MRB. IT IS USED TO CHAIN THE MRBS TOGETHER AND TO ***
***      PROVIDE THE FILE NAMES ASSOCIATED WITH THE MRB. THERE ***
***      WILL BE ONE MRB FOR EACH UNIQUE MAPNAME/Filename/SUSPENSE ***
***      FILENAME COMBINATION. THE FIRST BMRBH IS LOCATED VIA AN ***
***      ADDRESS POINTER IN THE OTB (OTBFMRBA).
***          ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|-----------------------------|
| 000000 | BMRBH DSECT | 17:14:05 08/15/85 |
| 000000 | BMRNEXTA DS A | ADDRESS OF NEXT BMRBH |
| 000004 | BMRFNAM DS 0CL10 | FILE NAME |
| 000004 | BMRDDNAM DS CL8 | DDNAME (OS) |
| 00000C | DS CL2 | UNUSED (OS) |
| 00000E | ORG BMRDDNAM | |
| 000004 | BMRDOSF DS CL7 | FILE NAME (DOS) |
| 00000B | BMRDOSYS DS CL3 | SYSNUM (DOS) |
| 00000E | BMRSFNFM DS 0CL10 | SUSPENSE FILE NAME |
| 00000E | BMRSFDD DS CL8 | SUSPENSE DDNAME (OS) |
| 000016 | DS CL2 | UNUSED (OS) |
| 000018 | ORG BMRSFDD | |
| 00000E | BMRSFDOS DS CL7 | SUSPENSE FILE NAME (DOS) |
| 000015 | BMRSFSYS DS CL3 | SUSPENSE FILE SYSNUM (DOS) |
| 000018 | BMRBLN DS H | LENGTH OF MRB FOLLOWING |
| | BMBNDI #FLAG X'80' | BIND THIS MAP FOR INPUT |
| 00001A | BMBNDII DS 0XL1 | |
| 00080 | BMBNDIM EQU X'80' | |
| | BMBNDO #FLAG X'40' | BIND THIS MAP FOR OUTPUT |
| 00001A | BMBNDOI DS 0XL1 | |
| 00040 | BMBNDOM EQU X'40' | |
| | BMBNDIO #FLAG X'20' | BIND THIS MAP AS AN I/O MAP |
| 00001A | BMBNDIOI DS 0XL1 | |
| 00020 | BMBNDIOM EQU X'20' | |
| 00001A | BMRBFL1 DS X | FLAG BYTE 1 |
| 00001B | DS X | RESERVED |
| 0001C | BMRLEN EQU *~-BMRBH | LENGTH OF BMRBH |

1.31 #BPADS

```

COPY #BPADS
*****
***          BPA: BATCH PARSE AREA (ADS)      ***
***          ***                                ***
***  BPA IS A DSECT THAT DESCRIBES THE WORK STORAGE AREAS USED  ***
***  BY THE PARSING MODULES IN ADS BATCH RUNTIME.      ***
***          ***                                ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|--|---|
| 000000 | BPA | DSECT | |
| | * | | |
| | * | BPA BLOCK 1 | |
| | * | | |
| | * | THE FOLLOWING VARIABLES ARE INTERFACE VARIABLES WITH THE | |
| | * | LEXICAL ANALYZER: | |
| | * | NXTTOKE, TOKEBUF, PACKED, BPANODEC, BPANODIG | |
| | * | | |
| | * | | |
| 000000 | BPAID | CL4 | 'BPA*' |
| 000008 | PDBL | DS | DS GENERALLY USEFUL DOUBLE WORD |
| 000010 | BPASCASV | DS | F SAVE AREA FOR STACK CHECKING |
| 000014 | BPASTKEA | DS | A STACK END ADDRESS |
| 000018 | BPASAVSZ | DS | H SIZE OF ENVIRONMENT SAVE AREA |
| 00001A | PHALF | DS | H GENERALLY USEFUL HALFWORD |
| 00001C | NXTTOKE | DS | H NEXT AVAILABLE SLOT FOR TOKEN |
| 00001E | TOKEBUF | DS | 2CL256 TOKEN BUFFER |
| 00200 | TOKEBLEN | EQU | *-TOKEBUF LENGTH OF TOKEN BUFFER |
| 00021E | INBUF | DS | CL80 TOKEN BUFFER |
| 00026E | PACKED | DS | XL9 HOLDING BUFFER FOR PACKED DEC # |
| 000277 | BPANODEC | DS | X # DECIMAL PLACES |
| 000278 | BPANODIG | DS | X # OF DIGITS IN DEC # |
| 000279 | BPAMBB | #FLAG | X'08' FIELD IS MULTIBIT BINARY |
| 000279 | BPAMBBI | DS | 0XL1 |
| 000008 | BPAMBBM | EQU | X'08' |
| 000279 | BPAEXP | DS | X FLAG BYTE FOR ADSOGEPR |
| 00027C | | DS | OF |
| | BPARC8 | #FLAG | X'80' RETURN CODE OF 8 IS ERROR |
| 00027C | BPARC8I | DS | 0XL1 |
| 000080 | BPARC8M | EQU | X'80' |
| | BPARC4 | #FLAG | X'40' RETURN CODE OF 4 IS WARNING |
| 00027C | BPARC4I | DS | 0XL1 |
| 000040 | BPARC4M | EQU | X'40' |
| | BPAPRMF | #FLAG | X'20' PARM EXPECTED IN ERR MSG ROUTINE |
| 00027C | BPAPRMFI | DS | 0XL1 |
| 000020 | BPAPRMFM | EQU | X'20' |
| 00027C | | DS | X FLAG BYTE |
| 00027D | | DS | X RESERVED |
| 000280 | | DS | 7F RESERVED |
| 0029C | BPALEN1 | EQU | *-BPA FIRST BPA BLOCK SIZE |
| | * | | |
| | * | BPA BLOCK 2 | |
| | * | | |
| | * | | |
| | * | THE FOLLOWING VARIABLES ARE APPLICABLE ONLY TO THE | |
| | * | PARSE MODULES. | |
| | * | | |
| 00029C | BPABLK3 | DS | 0F |
| 00029C | BPAWPNUM | DS | F SAVE AREA FOR NUMBER |
| 0002A0 | BPAPIST | DS | 5D DC MACRO PARM AREA |

| | | | | |
|--------|--|-------|------------|------------------------------------|
| 0002C8 | BPADLGNM | DS | CL8 | CONTROL CMD DLG/PGM NAME (LITERAL) |
| 0002D0 | BPAWKCL8 | DS | CL8 | TEMPORARY WORK AREA FOR DLG NAME |
| 0002D8 | BPASAVRG | DS | F | TEMP REGISTER STORAGE AREA |
| 0002DC | BPAPATCH | DS | A | ADDRESS OF ADSORUNB PATCH AREA |
| 0002E0 | BPARPTBA | DS | A | ADDR OF PRINT BUFFER |
| 0002E4 | LEXADDR | DS | A | ADDR OF CURRENT TOKEN START |
| 0002E8 | NXTNUMB | DS | H | NEXT AVAILABLE SLOT IN NUMBER |
| 0002EA | LEXLNTH | DS | H'0' | LNTH OF STRING RETURNED FROM L.A. |
| 0002EC | LEXCOLM | DS | H'72' | STARTING COLUMN OF TOKEN |
| 0002EE | FSTCOLM | DS | H'01' | FIRST COLUMN TO BE USED FOR INPUT |
| 0002F0 | LSTCOLM | DS | H'72' | LAST COLUMN TO BE USED FOR INPUT |
| 0002F4 | NUMBER | DS | 10F | NUMBER TOKEN LIST |
| | * DLRBUFF, MSGCBUFF, AND SRCBUFF ARE BUILD AREAS FOR ENTRIES | | | |
| | * INTO THE PRINT BUFFER FOR COMPILED PROCESS SOURCE | | | |
| 00031C | DLRBUFF | DS | OF | BUILD AREA FOR \$ LINE ENTRY |
| | DLRFLG | #FLAG | X'40' | INDICATES \$ LINE ENTRY |
| 00031C | DLRFLGI | DS | 0XL1 | |
| 00040 | DLRFLGM | EQU | X'40' | |
| 00031C | DLRFLGB | DS | X | DOLLAR FLAG BYTE |
| 00031D | DLRNUM | DS | X | NUMBER OF \$'S IN ENTRY |
| 00002 | DLRHDRSZ | EQU | *-DLRBUFF | SIZE OF \$ ENTRY HEADER |
| 00031E | DLRCOL | DS | 10X | COLUMN #'S FOR \$'S |
| 0000A | DLRMAX | EQU | *-DLRCOL | MAX # OF ENTRIES IN DLRBUFF |
| 000328 | MSGCBUFF | DS | OF | BUILD AREA FOR MSG CODE ENTRY |
| | MSGCFLG | #FLAG | X'20' | INDICATES MSG CODE ENTRY |
| 000328 | MSGCFGLI | DS | 0XL1 | |
| 00020 | MSGCFGLM | EQU | X'20' | |
| 000328 | MSGCFLGB | DS | X | MESSAGE CODE FLAG BYTE |
| 000329 | MSGCNUM | DS | X | NUMBER OF MSG CODES IN ENTRY |
| 00002 | MSGHDRSZ | EQU | *-MSGCBUFF | |
| 00032A | MSGCODES | DS | 40X | MSG CODE NUMBERS |
| 000354 | SRCBUFF | DS | OF | BUILD AREA FOR SOURCE ENTRY |
| | SRCFLG | #FLAG | X'80' | INDICATES SOURCE ENTRY |
| 000354 | SRCFLGI | DS | 0XL1 | |
| 00080 | SRCFLGM | EQU | X'80' | |
| 000354 | SRCFLGB | DS | X | SOURCE FLAG BYTE |
| 000355 | SRCSEQNO | DS | 7X | SOURCE SEQUENCE NUMBER |
| | FRSTCRD | #FLAG | X'80' | ON IF NO PROCESS CARD READ YET |
| 00035C | FRSTCRDI | DS | 0XL1 | |
| 00080 | FRSTCRDM | EQU | X'80' | |
| | SRCERR | #FLAG | X'40' | ON IF CURRENT CARD HAS ERROR |
| 00035C | SRCERRI | DS | 0XL1 | |
| 00040 | SRCERRM | EQU | X'40' | |
| | PRINTED | #FLAG | X'20' | ON IF CARD IMAGE NOT TO BE PRNTED |
| 00035C | PRINTEDI | DS | 0XL1 | |
| 00020 | PRINTEDM | EQU | X'20' | |
| | MXLERR | #FLAG | X'10' | ON IF MAX # ERRS FOR LINE FOUND |
| 00035C | MXLERRI | DS | 0XL1 | |
| 00010 | MXLERRM | EQU | X'10' | |
| 00035C | BPAPRBFL | DS | X | PRINT BUFF FLAG BYTE |
| | FAIL | #FLAG | X'80' | ON IF TOKEN FAILS TEST |
| 00035D | FAILI | DS | 0XL1 | |
| 00080 | FAILM | EQU | X'80' | |
| | EOF | #FLAG | X'40' | ON IF SOURCE INPUT COMPLETE |
| 00035D | EOFI | DS | 0XL1 | |
| 00040 | EOFM | EQU | X'40' | |
| 00035D | BPAFLAG1 | DS | X | FIRST FLAG BYTE |
| 00035E | BPADATYP | DS | X | IRA DATA TYPE |
| 00035F | DELIM | DS | C'*' | COMMENT DELIMITER CHARACTER |
| 000360 | HOLDCC | DS | CL1 | |
| 000361 | ZONEDNO | DS | XL16 | HOLDING BUFFER FOR ZONED DEC # |
| 000371 | BPAOPCOD | DS | X | OP CODE FOR PROCEDURAL CME |
| 000372 | BRPLDSRG | DS | C | DSORG OF BRPL |
| 000374 | LEXLNTH2 | DS | H | LEXLNTH SAVE AREA FOR STR/BIN STR |
| | LEXSTR | #FLAG | X'80' | STRING CONSTANT ALREADY PARSED |
| 000376 | LEXSTRI | DS | 0XL1 | |

| | | | |
|--------|-------|---|--------------------------------------|
| | 00080 | LEXSTRM EQU X'80' | |
| | | LEXBIN #FLAG X'40' | BINARY STRING CONSTANT IS PARSED |
| 000376 | | LEXBINI DS OXL1 | |
| | 00040 | LEXBINM EQU X'40' | |
| | | LEXFLTS #FLAG X'20' | SHORT FLOAT CONSTANT PARSE*LCB84229* |
| 000376 | | LEXFLTSI DS OXL1 | |
| | 00020 | LEXFLTSM EQU X'20' | |
| | | LEXFLTL #FLAG X'10' | LONG FLOAT CONSTANT PARSED*LCB84229* |
| 000376 | | LEXFLTTLI DS OXL1 | |
| | 00010 | LEXFLTLM EQU X'10' | |
| 000376 | | LEXFLAG1 DS X | FLAG BYTE FOR STR/BIN CONSTANTS |
| 000377 | | BPABINL DS X | LENGTH OF MULTIBIT BINARY FIELD |
| 000378 | | BPABIN DS F | MULTIBIT BINARY VALUE |
| 00037C | | BRPLINA DS A | ADDR OF INPUT BRPL |
| 000380 | | BRPLOUTA DS A | ADDR OF OUTPUT BRPL |
| 000384 | | BRPLLEN DS H | LENGTH OF BRPL |
| 000386 | | LEXCOLM2 DS H | LEXCOLM SAVE AREA FOR STR/BIN STR |
| | 00388 | BPAFLEN EQU ((*-BPA+3)/4)*4 | BPA FIXED LENGTH PORTION |
| | | *** THE FOLLOWING STORAGE IS USED BY THE BATCH GENERATOR PARSER *** | |
| 000388 | | BPAIOPRM DS OF | PARMS FOR IDMSUTIO |
| 000388 | | BPAIOP1 DS A | |
| 00038C | | BPAIOP2 DS A | |
| 000390 | | BPAIOP3 DS A | |
| 000394 | | BPAIOP4 DS A | |
| 000398 | | BPAFUNC DS CL4 | FUNCTION CODE |
| 00039C | | BPASTAT DS CL4 | STATUS, RETURN CODE, OR ABEND CODE |
| 0003A0 | | BPAFILE DS CL4 | FILE NAME |
| 0003A4 | | BPAUTIOA DS A | ADDR OF IDMSUTIO |
| 0003A8 | | BPAPRTLA DS A | ADDR OF PRINT LINE ROUTINE |
| 0003AC | | BPAPEERRA DS A | ADDRESS OF ERROR PRINT ROUTINE |
| | | BPARDIN #FLAG X'80' | ON IF READ INPUT FROM SYSIPT |
| 0003B0 | | BPARDINI DS OXL1 | |
| | 00080 | BPARDINM EQU X'80' | |
| | | BPALIST #FLAG X'40' | ON IF WRITE OUTPUT TO SYSOUT |
| 0003B0 | | BPALISTI DS OXL1 | |
| | 00040 | BPALISTM EQU X'40' | |
| | | BPANOPR #FLAG X'20' | ON IF LINE ALREADY PRINTED |
| 0003B0 | | BPANOPRI DS OXL1 | |
| | 00020 | BPANOPRM EQU X'20' | |
| | | SEQCHK #FLAG X'10' | ON IF SEQUENCE CHECKING IN EFFECT |
| 0003B0 | | SEQCHKI DS OXL1 | |
| | 00010 | SEQCHKM EQU X'10' | |
| | | *BPASFNF #FLAG X'08' | ON IF SYSIPT FILE NOT FND*RQE86104* |
| 0003B0 | | BPASFNF1 DS OXL1 | ON IF SYSIPT FILE NOT FND*RQE86104* |
| | 00008 | BPASFNFM EQU X'08' | |
| 0003B0 | | DS X | READ INPUT/WRITE OUTPUT FLAG |
| 0003B4 | | DS OF | |
| 0003B4 | | BPAERCOD DS F | ERROR CODE FOR WTL |
| 0003B8 | | BPAERTRN DS A | RETURN REG FOR BATCH GENERATOR |
| 0003BC | | PRNTHR12 DS F | SAVE REG AREA FOR PRINT ROUTINE |
| 0003C0 | | BPALNLMT DS H | TOTAL LINES PER OUTPUT PAGE |
| 0003C2 | | BPALNCNT DS H | LINE COUNT FOR PAGE |
| 0003C4 | | BPAPGCNT DS H | PAGE COUNTER |
| 0003C6 | | BPAJULDT DS PL3 | JULIAN DATE (YYDDD+ FORMAT) PACKED |
| 0003C9 | | BPAGRGDT DS PL4 | GREGORIAN DATE (0YYMMDD+ FORMAT) |
| 0003CD | | BPATIME DS PL4 | TIME (0HHMMSS+ FORMAT) PACKED |
| 0003D1 | | BPAgreg DS CL6 | GREG DATE (YYMMDD FORMAT) UNPACKED |
| 0003D7 | | BPATME DS CL6 | TIME (HHMMSS FORMAT) UNPACKED |
| | | *** THE FOLLOWING ARE PRINT LINE LAYOUTS | |
| 0003E0 | | DS OF | |
| 0003E0 | | BPAPRTLN DS OCL133 | PRINT LINE BUFFER AREA |
| 0003E0 | | DS C' ' | CARRIAGE CONTROL CHARACTER |
| 0003E1 | | BPAbegln DS CL2 | START OF PRINT LINE |
| 0003E3 | | BPASTMT# DS CL5 | STATEMENT NUMBER |
| 0003E8 | | DS CL7 | |

| | | | | |
|--------|----------|-----|-----------------|------------------------------------|
| 0003EF | BPAINTXT | DS | CL80 | INPUT STATEMENT TEXT |
| 00043F | | DS | CL28 | |
| 00045B | | ORG | BPABEGLN | INPUT MESSAGE LINE LAYOUT |
| 0003E1 | | DS | CL10 | |
| 0003EB | BPAEFLAG | DS | CL3 | ERROR MESSAGE INDICATOR '***' |
| 0003EE | | DS | CL8 | |
| 0003F6 | BPAMTEXT | DS | CL80 | MESSAGE TEXT AREA |
| 000446 | | DS | CL32 | |
| 000466 | | ORG | BPAMTEXT | ERROR MESSAGE AREA |
| 0003F6 | BPAPRCOD | DS | CL6 | ERROR MESSAGE CODE |
| 0003FC | | DS | CL3 | |
| 0003FF | BPAETEXT | DS | CL80 | ERROR MESSAGE TEXT AREA |
| 00044F | | DS | CL24 | |
| 000467 | | ORG | BPAETEXT | PROCESS STATEMENT PRINT AREA |
| 0003FF | | DS | CL5 | |
| 000404 | BPAPRTXT | DS | CL80 | DIALOG PROCESS PRINT AREA |
| 000454 | | DS | CL19 | |
| 000467 | | ORG | , | *DKJ86008* |
| 000467 | BPAPRHD2 | DS | 0CL133 | VARIABLE PAGE HEADER *DKJ86008* |
| 000467 | | DS | C' ' | *DKJ86008* |
| 000468 | BPATAPE | DS | CL6 | TAPE # |
| 00046E | | DS | CL42 | |
| 000498 | BPAPLICM | DS | CL32 | LICENSE MESSAGE |
| 0004B8 | | DS | CL30 | |
| 0004D6 | BPAPDATE | DS | CL8 | |
| 0004DE | | DS | CL1' ' | |
| 0004DF | BPAPTIME | DS | CL6 | |
| 0004E5 | | DS | CL1' ' | |
| 0004E6 | BPAPPAGE | DS | CL5 | |
| 0004EB | | DS | CL1 | *DKJ86008* |
| 0004EC | SEQSTRT | DS | H | START COLUMN FOR SEQUENCE NUMBER |
| 0004EE | SEQLNTH | DS | H | LENGTH OF SEQUENCE NUMBER |
| 0004F0 | PREVSEQ | DS | CL80 | SEQUENCE NUMBER FROM PREVIOUS CARD |
| 000540 | BPACNODA | DS | A | ADDRESS OF CURRENT PARSE TREE NODE |
| 000544 | BPABR13 | DS | F | BATCH REGENERATOR'S R13 ON ENTRY |
| 000548 | BPAREGSV | DS | 16F | MULTIPLE REGISTER SAVE AREA |
| 000588 | BPAOTBA | DS | A | ADDR OF OTB |
| 0058C | BPALEN | EQU | ((*-BPA+3)/4)*4 | LENGTH OF FIXED PORTION OF BPA |
| 008AC | BPATLEN | EQU | BPALEN+800 | LENGTH OF BPA PLUS STACK SPACE |
| 00058C | BPASTACK | DS | 0F | START OF PARSER STACK |

1.32 #BPXDS

```

COPY #BPXDS
*****
* ---FILE MAPPING BUFFER PREFIX---
*
* THIS PREFIX PRECEDES EACH BUFFER ACQUIRED BY THE FILE MAPPING
* RUNTIME SYSTEM.
*
* FUNCTION: MAINTAIN CONTROL INFORMATION FOR:
*
*      1) BUFFER CHAIN
*      2) DEBLOCKING/BLOCKING
*      3) I/O STATUS
*
* DATASET BUFFERS ARE LINKED INTO A CIRCULAR CHAIN TO FACILITATE
* MANIPULATION BY THE I/O SUBTASK.
*
* THE ADDRESS OF THE DATASET BUFFER AREA (THE STORAGE ELEMENT WHICH
* IS PARTITIONED INTO BUFFERS) IS MAINTAINED IN BLEFIOBA.
*
* THE LENGTH OF AN ALLOCATED BPX VARIES ACCORDING TO THE OPERATING
* SYSTEM AND ACCESS METHOD; IN FACT, THE BPX CONTAINS
* OPERATING SYSTEM/ACCESS METHOD CONTROL BLOCKS USED BY THE
* OPERATING SYSTEM/ACCESS METHOD DEPENDENT COMPONENTS OF FILE
* MAPPING. IN GENERAL, THE BPX IS COMPOSED OF THE FOLLOWING:
*
*      1) THE BUFFER PREFIX ELEMENT ITSELF, AS DESCRIBED BY THE
*         DSECT                                     09/27/90
*      2) THE OPERATING SYSTEM/ACCESS METHOD REQUEST CONTROL ELEMENT,
*         NAMELY THE NON-VSAM DECB OR VSAM RPL
*      3) A 4-BYTE LENGTH FIELD, CONTAINING THE LENGTH THE
*         PHYSICAL RECORD MAINTAINED IN THE BUFFER
*
* THE LENGTH OF THE BPX IS THE SUM OF THE LENGTHS OF ALL OF THE
* ABOVE.
* THE PHYSICAL RECORD BEGINS IMMEDIATELY AFTER THE BPX.
*
* A LIST OF BUFFER PREFIXES
*
* OS NON-VSAM: BPXDSLEN + L'DECB + PHYSICAL RECORD LEN FIELD
* DOS NON-VSAM: BPXDSLEN + PHYSICAL RECORD LEN FIELD
* OS/DOS VSAM: BPXDSLEN + L'RPL + PHYSICAL RECORD LEN FIELD
*
* ---BUFFER ELEMENT CHAINING---
*
* A BUFFER ELEMENT MAY EXIST WITHIN TWO CHAINS:
*
* 1) THE BLE I/O BUFFER CHAIN, WHICH CONTAINS ALL BUFFERS FOR THE
*    DATASET DESCRIBED BY THE BLE.
*
*    BUFFER ELEMENTS ARE INSERTED INTO THIS CHAIN WHEN THE DATASET
*    IS OPENED, AND DELETED FROM THE CHAIN WHEN THE DATASET IS
*    CLOSED.
*
* 2) THE SUBTASK I/O BUFFER CHAIN, WHICH CONTAINS ALL BUFFERS
*    FOR WHICH PHYSICAL I/O IS IN PROGRESS.
*
*    BUFFER ELEMENTS ARE INSERTED INTO THIS CHAIN WHEN PHYSICAL I/O
*    IS REQUIRED, AND DELETED FROM THIS CHAIN WHEN THE PHYSICAL I/O
*    HAS COMPLETED.
*
* ---BUFFER PREFIX FLAG SETTINGS---

```

```

*
*   BPXRDY (READY FLAG) - BUFFER HAS BEEN COMPLETELY FILLED (OUTPUT)
*   OR EMPTIED (INPUT) AND IS READY FOR
*   PROCESSING BY THE I/O SUBTASK.
*
*
*   BPXBUSY (BUSY FLAG) - BUFFER HAS BEEN ACCEPTED BY THE SUBTASK
*   FOR PROCESSING
*
*
*   BUSY=0 + READY=0      - BUFFER I/O SERVICES (RHDCBIOS) CAN PERFORM
*                           LOGICAL I/O FOR THIS BUFFER; NO ACCESS BY
*                           SUBTASK ALLOWED.
*
*   BUSY=1 + READY=0      - SUBTASK HAS ACCEPTED THIS BUFFER; NO
*                           ACCESS BY BUFFER I/O SERVICES ALLOWED.
*
*   BUSY=0 + READY=1      - BUFFER IS AVAILABLE TO SUBTASK; NO ACCESS
*                           BY BUFFER I/O SERVICES ALLOWED.
*
*   BUSY=1 + READY=1      - CANNOT OCCUR
*
*
*   BPXWAIT (WAIT FLAG) - SUBTASK HAS REQUESTED SERVICES FOR THE
*                           BUFFER WHICH HAVE RESULTED IN AN OPERATING
*                           SYSTEM WAIT FOR THE I/O TO COMPLETE.
*
*
*   BPXLIOP (LOGICAL I/O FLAG) - LOGICAL I/O IS IN PROGRESS FOR THIS
*                           BUFFER
*
*   BPXIOP (PHYSICAL I/O FLAG) - PHYSICAL I/O IS IN PROGRESS FOR
*                           THIS BUFFER
*
*****

```

Offset Value

| | | | |
|--------|----------|------------------------------|------------------------------------|
| 000000 | BPX | DSECT | 14:51:54 03/13/87 |
| 000000 | BPXID | DS CL4 | 'BPX*' EYECATCHER |
| 000004 | BPXPRVA | DS A | A(PREV BPX) |
| 000008 | BPXNXTA | DS A | A(NEXT BPX) |
| 00000C | BPXLEN | DS H | LENGTH OF THIS BPX |
| 00000E | | DS H | RESERVED |
| | * | | |
| | BPXIOP | #FLAG X'80' | BUFFER PHYSICAL I/O IN PROGRESS |
| 000010 | BPXPIOPI | DS 0XL1 | |
| | 000080 | BPXPIOPM EQU X'80' | |
| | BPXLIOP | #FLAG X'40' | BUFFER LOGICAL I/O IN PROGRESS |
| 000010 | BPXLIOP1 | DS 0XL1 | |
| | 000040 | BPXLIOPM EQU X'40' | |
| | BPXRDY | #FLAG X'20' | BUFFER AVAILABLE TO SUBTASK |
| 000010 | BPXRDYI | DS 0XL1 | |
| | 000020 | BPXRDYM EQU X'20' | |
| | BPXBUSY | #FLAG X'10' | BUFFER ACCEPTED BY SUBTASK |
| 000010 | BPXBUSYI | DS 0XL1 | |
| | 000010 | BPXBUSYM EQU X'10' | |
| | BPXINV | #FLAG BPXRDYM+BPXBUSYM | BUFR UNAVAILABLE TO BUFR I/O SERV |
| 000010 | BPXINVI | DS 0XL1 | |
| | 000030 | BPXINVM EQU BPXRDYM+BPXBUSYM | |
| | BPXWAIT | #FLAG X'08' | SUBTASK WAITING FOR I/O COMPLETION |
| 000010 | BPXWAITI | DS 0XL1 | |
| | 000008 | BPXWAITM EQU X'08' | |
| | BPXMRL | #FLAG X'04' | MODIFY RECORD LENGTH BEFORE I/O |
| 000010 | BPXMRLI | DS 0XL1 | |

| | | | | | |
|--------|-------|-----------|-------|-------|-----------------------------------|
| | 00004 | BPXMRLM | EQU | X'04' | |
| 000010 | | BPXFLAG1 | DS | X | FLAG BYTE 1 |
| | * | | | | |
| | | BPXEOF | #FLAG | X'80' | END-OF-FILE REACHED FOR DATASET |
| 000011 | | BPXEOF1 | DS | 0XL1 | |
| | 00080 | BPXEOFM | EQU | X'80' | |
| | | BPXSYND | #FLAG | X'40' | SYNAD EXIT WAS TAKEN FOR DATASET |
| 000011 | | BPXSYNDI | DS | 0XL1 | |
| | 00040 | BPXSYNDM | EQU | X'40' | |
| | | BPXIOER | #FLAG | X'20' | I/O ERROR OCCURED (DOS) |
| 000011 | | BPXIOERI | DS | 0XL1 | |
| | 00020 | BPXIOERM | EQU | X'20' | |
| 000011 | | BPXFLAG2 | DS | X | FLAG BYTE 2 |
| | * | | | | |
| | | BPXCCR | #FLAG | X'40' | RECFM ERROR IN CONCATENATION |
| 000012 | | BPXCCRI | DS | 0XL1 | |
| | 00040 | BPXCCRM | EQU | X'40' | |
| | | BPXCCB | #FLAG | X'20' | BLKSIZE ERROR IN CONCATENATION |
| 000012 | | BPXCCBI | DS | 0XL1 | |
| | 00020 | BPXCCBM | EQU | X'20' | |
| | | BPXCLL | #FLAG | X'10' | LRECL ERROR IN CONCATENATION |
| 000012 | | BPXCLLI | DS | 0XL1 | |
| | 00010 | BPXCLLM | EQU | X'10' | |
| | | BPXR RD | #FLAG | X'08' | REISSUE READ FOR THIS BUFFER |
| 000012 | | BPXR RD1 | DS | 0XL1 | |
| | 00008 | BPXR RD M | EQU | X'08' | |
| 000012 | | BPXFLAG3 | DS | X | FLAG BYTE 3 |
| | * | | | | |
| | | BPXINP | #FLAG | X'80' | PERFORM INPUT OPERATION |
| 000013 | | BPXINPI | DS | 0XL1 | |
| | 00080 | BPXINPM | EQU | X'80' | |
| | | BPXOUT | #FLAG | X'40' | PERFORM OUTPUT OPERATION |
| 000013 | | BPXOUTI | DS | 0XL1 | |
| | 00040 | BPXOUTM | EQU | X'40' | |
| 000013 | | BPXFLAG4 | DS | X | FLAG BYTE 4 |
| | * | | | | |
| 000014 | | BPXNPOS | DS | F | DISP TO LGCL REC WITHIN PHYS REC |
| 000018 | | BPXLOC | DS | F | LOCATION OF PHYS RECORD ON DEVICE |
| 00001C | | BPXBLEA | DS | A | A(BLE OWNING THIS BUFFER ELEMENT) |
| 000020 | | BPXIONXA | DS | A | A(NEXT BUFR ELEMENT IN I/O CHAIN) |
| 000024 | | BPXD FR15 | DS | F | R15 R/C FROM DOS FILE I/O |
| 000028 | | BPXMSG# | DS | PL4 | ERROR MESSAGE NUMBER |
| 00002C | | | DS | F | RESERVED |
| | 00030 | BPXDSLEN | EQU | *-BPX | LENGTH OF BPX FIXED PORTION |
| 000030 | | BPXAMRB | DS | 0F | A(DECB RPL) |

1.33 #BRPLDS

```

COPY #BRPLDS
*****
*   BATCH MAPPING REQUEST PARAMETER LIST
*
*   COMMUNICATES INFORMATION CONCERNING THE CHARACTERISTICS OF THE
*   DATASET TO BE PROCESSED AND THE SERVICE REQUIRED FROM BATCH
*   MAPPING.
*
*       1) #MREQ - THE INFORMATION COMMUNICATED HAS BEEN MAINTAINED IN
*          THE MAP LOAD MODULE AND IN THE #MREQ PLIST
*
*       2) DIRECT-CALL - THE INFORMATION COMMUNICATED IS SUPPLIED BY
*          THE PROGRAM ISSUING THE DIRECT-CALL MACROS
*
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|--------------------------------------|
| 000000 | BRPL DSECT | 10:20:00 07/07/86 |
| 000000 | BRPLID DS CL4 | EYECATCHER C'BRPL' |
| 000004 | BRPLNXTA DS A | A(NEXT BRPL) |
| 000008 | BRPLREQ DS X | REQUEST FLAGS |
| 000009 | 00001 BRPLOPEN EQU 1 | OPEN REQUEST |
| 000009 | 00002 BRPLCLOS EQU 2 | CLOSE REQUEST |
| 000009 | 00003 BRPLREAD EQU 3 | READ REQUEST |
| 000009 | 00004 BRPLWRIT EQU 4 | WRITE REQUEST |
| 000009 | 00005 BRPLCKPT EQU 5 | CHECKPOINT REQUEST |
| 000009 | 00006 BRPLRSTR EQU 6 | RESTART REQUEST) |
| 000009 | 00007 BRPLGEN EQU 7 | GENERATE BRPL REQUEST |
| 000009 | 00008 BRPLSET EQU 8 | SET BRPL REQUEST |
| 000009 | 00009 BRPLSHOW EQU 9 | SHOW BRPL REQUEST |
| 000009 | 0000A BRPLLREQ EQU 10 | BRPL LENGTH REQUEST |
| 000009 | 0000B BRPLPS #FLAG X'80' | PHYSICAL SEQUENTIAL |
| 000009 | 0000C BRPLPSI DS 0XL1 | |
| 000009 | 0000D BRPLPSM EQU X'80' | |
| 000009 | 0000E BRPLSI #FLAG X'40' | SYSIN (OS) (IMPLIES INPUT) |
| 000009 | 0000F BRPLSII DS 0XL1 | |
| 000009 | 00010 BRPLSIM EQU X'40' | |
| 000009 | 00011 BRPLCD #FLAG X'20' | CARD (DOS) |
| 000009 | 00012 BRPLCDI DS 0XL1 | |
| 000009 | 00013 BRPLCDM EQU X'20' | |
| 000009 | 00014 BRPLSO #FLAG X'10' | SYSOUT (OS) (IMPLIES OUTPUT) |
| 000009 | 00015 BRPLSOI DS 0XL1 | |
| 000009 | 00016 BRPLSOM EQU X'10' | |
| 000009 | 00017 BRPLPR #FLAG X'08' | PRINT (DOS) (IMPLIES OUTPUT) |
| 000009 | 00018 BRPLPRI DS 0XL1 | |
| 000009 | 00019 BRPLPRM EQU X'08' | |
| 000009 | 00020 BRPLKS #FLAG X'04' | RESERVED FOR VSAM KSDS (UNSUPPORTED) |
| 000009 | 00021 BRPLKSI DS 0XL1 | |
| 000009 | 00022 BRPLKSM EQU X'04' | |
| 000009 | 00023 BRPLES #FLAG X'02' | VSAM ESDS |
| 000009 | 00024 BRPLESI DS 0XL1 | |
| 000009 | 00025 BRPLESM EQU X'02' | |
| 000009 | 00026 BRPLRR #FLAG X'01' | RESERVED FOR VSAM RRDS (UNSUPPORTED) |
| 000009 | 00027 BRPLRII DS 0XL1 | |
| 000009 | 00028 BRPLRRM EQU X'01' | |
| 000009 | 00029 BRPLVS #FLAG BRPLKSM+BRPLESM+BRPLRRM VSAM DATASET | |
| 000009 | 00030 BRPLVSI DS 0XL1 | |
| 000009 | 00031 BRPLVSM EQU BRPLKSM+BRPLESM+BRPLRRM | |
| 000009 | 00032 BRPLDSO DS X TYPE OF DATASET ORGANIZATION BYTE | |
| | * | |

| | | | |
|--------|-------|--|---------------------------------|
| | | BRPLEOF #FLAG X'80' | END-OF-FILE REACHED |
| 00000A | 00080 | BRPLEOFI DS OXL1 | |
| | | BRPLEOFM EQU X'80' | |
| | | BRPLSYN #FLAG X'40' | PHYSICAL I/O ERROR OCCURRED |
| 00000A | 00040 | BRPLSYNI DS OXL1 | |
| | | BRPLSYNM EQU X'40' | |
| | | BRPLTMP #FLAG X'20' | TEMPORARY DATASET |
| 00000A | 00020 | BRPLTMI DS OXL1 | |
| | | BRPLTMPM EQU X'20' | |
| | | BRPLPDS #FLAG X'10' | MEMBER OF PARTITIONED DATASET |
| 00000A | 00010 | BRPLPDSI DS OXL1 | |
| | | BRPLPDSM EQU X'10' | |
| | | BRPLDMY #FLAG X'08' | DUMMY DATASET |
| 00000A | 00008 | BRPLDMYI DS OXL1 | |
| | | BRPLDMYM EQU X'08' | |
| | | BRPLGU #FLAG X'04' | GET-UPDATE PERFORMED (VSAM I/O) |
| 00000A | 00004 | BRPLGUI DS OXL1 | |
| | | BRPLGUM EQU X'04' | |
| 00000A | | BRPLFST DS X | FILE STATUS BYTE |
| | * | | |
| | | BRPLINP #FLAG X'80' | DATASET OPENED INPUT |
| 00000B | 00080 | BRPLINPI DS OXL1 | |
| | | BRPLINPM EQU X'80' | |
| | | BRPLOUT #FLAG X'40' | DATASET OPENED OUTPUT |
| 00000B | 00040 | BRPLOUTI DS OXL1 | |
| | | BRPLOUTM EQU X'40' | |
| | | BRPLIO #FLAG BRPLINPM+BRPLOUTM | DATASET OPENED INPUT+OUTPUT |
| 00000B | 000C0 | BRPLIOI DS OXL1 | |
| 00000B | | BRPLIOM EQU BRPLINPM+BRPLOUTM | |
| | | BRPLUSAG DS X | DATASET USAGE |
| | * | | |
| | | BRPLBLK #FLAG X'10' | BLOCKED RECORDS |
| 00000C | 00010 | BRPLBLKI DS OXL1 | |
| | | BRPLBLKM EQU X'10' | |
| | | BRPLASA #FLAG X'04' | ASA (ANSI) CARRIAGE CTL |
| 00000C | 00004 | BRPLASAI DS OXL1 | |
| | | BRPLASAM EQU X'04' | |
| | | BRPLMAC #FLAG X'02' | MACHINE CARRIAGE CTL |
| 00000C | 00002 | BRPLMACI DS OXL1 | |
| | | BRPLMACM EQU X'02' | |
| 00000C | | BRPLCC #FLAG BRPLASAM+BRPLMACM | CARRIAGE CONTROL EXISTS |
| 00000C | 00006 | BRPLCCI DS OXL1 | |
| | | BRPLCCM EQU BRPLASAM+BRPLMACM | |
| | * | | |
| | | BRPLF #FLAG X'80' | FIXED |
| 00000C | 00080 | BRPLFI DS OXL1 | |
| | | BRPLFM EQU X'80' | |
| | | BRPLFA #FLAG BRPLFM+BRPLASAM | FIXED ASA |
| 00000C | 00084 | BRPLFAI DS OXL1 | |
| | | BRPLFAM EQU BRPLFM+BRPLASAM | |
| | | BRPLFM #FLAG BRPLFM+BRPLMACM | FIXED MACH |
| 00000C | 00082 | BRPLFMI DS OXL1 | |
| | | BRPLFMM EQU BRPLFM+BRPLMACM | |
| | | BRPLFB #FLAG BRPLFM+BRPLBLKM | FIXED BLKED |
| 00000C | 00090 | BRPLFBII DS OXL1 | |
| | | BRPLFBM EQU BRPLFM+BRPLBLKM | |
| | | BRPLFBA #FLAG BRPLFM+BRPLBLKM+BRPLASAM | FIXED BLKED ASA |
| 00000C | 00094 | BRPLFBAM EQU BRPLFM+BRPLBLKM+BRPLASAM | |
| | | BRPLFBM #FLAG BRPLFM+BRPLBLKM+BRPLMACM | FIXED BLKED MACH |
| 00000C | 00092 | BRPLFBMI DS OXL1 | |
| | | BRPLFBMM EQU BRPLFM+BRPLBLKM+BRPLMACM | |
| | * | | |
| | | BRPLV #FLAG X'40' | VARIABLE |
| 00000C | 00040 | BRPLVI DS OXL1 | |
| | | BRPLVM EQU X'40' | |

| | | | |
|--------|----------------|--|---|
| | BRPLVA | #FLAG BRPLVM+BRPLASAM | VAR ASA |
| 00000C | BRPLVAI | DS 0XL1 | |
| | 00044 BRPLVAM | EQU BRPLVM+BRPLASAM | |
| | BRPLVM | #FLAG BRPLVM+BRPLMACM | VAR MACH |
| 00000C | BRPLVMI | DS 0XL1 | |
| | 00042 BRPLVMM | EQU BRPLVM+BRPLMACM | |
| | BRPLVB | #FLAG BRPLVM+BRPLBLKM | VAR BLKED |
| 00000C | BRPLVBI | DS 0XL1 | |
| | 00050 BRPLVBM | EQU BRPLVM+BRPLBLKM | |
| | BRPLVBA | #FLAG BRPLVM+BRPLBLKM+BRPLASAM | VAR BLKED ASA |
| 00000C | BRPLVBAI | DS 0XL1 | |
| | 00054 BRPLVBAM | EQU BRPLVM+BRPLBLKM+BRPLASAM | |
| | BRPLVBM | #FLAG BRPLVM+BRPLBLKM+BRPLMACM | VAR BLKED MACH |
| 00000C | BRPLVBMI | DS 0XL1 | |
| | 00052 BRPLVBMM | EQU BRPLVM+BRPLBLKM+BRPLMACM | |
| | * | | |
| | BRPLU | #FLAG BRPLFM+BRPLVM | UNDEFINED |
| 00000C | BRPLUI | DS 0XL1 | |
| | 000C0 BRPLUM | EQU BRPLFM+BRPLVM | |
| | BRPLUA | #FLAG BRPLFM+BRPLVM+BRPLASAM | UND ASA |
| 00000C | BRPLUAI | DS 0XL1 | |
| | 000C4 BRPLUAM | EQU BRPLFM+BRPLVM+BRPLASAM | |
| | BRPLUM | #FLAG BRPLFM+BRPLVM+BRPLMACM | UND MACH |
| 00000C | BRPLUMI | DS 0XL1 | |
| | 000C2 BRPLUMM | EQU BRPLFM+BRPLVM+BRPLMACM | |
| 00000C | BRPLRCFM | DS X RECORD FORMAT | |
| | * | | |
| | BRPLAL | #FLAG X'40' | ANSI LABEL (OS) |
| 00000D | BRPLALI | DS 0XL1 | |
| | 00040 BRPLALM | EQU X'40' | |
| | BRPLAUL | #FLAG X'44' | ANSI + USER LABEL (OS) |
| 00000D | BRPLAULI | DS 0XL1 | |
| | 00044 BRPLAULM | EQU X'44' | |
| | BRPLBLP | #FLAG X'10' | LABEL PROCESSING BYPASSED (OS) |
| 00000D | BRPLBLPI | DS 0XL1 | |
| | 00010 BRPLBLPM | EQU X'10' | |
| | BRPLSUL | #FLAG X'0A' | STANDARD + USER LABEL |
| 00000D | BRPLSULI | DS 0XL1 | |
| | 0000A BRPLSULM | EQU X'0A' | |
| | BRPLNSL | #FLAG X'04' | NON-STANDARD LABEL |
| 00000D | BRPLNSLI | DS 0XL1 | |
| | 00004 BRPLNSLM | EQU X'04' | |
| | BRPLSL | #FLAG X'02' | STANDARD LABEL |
| 00000D | BRPLSLI | DS 0XL1 | |
| | 00002 BRPLSLM | EQU X'02' | |
| | BRPLNL | #FLAG X'01' | NO LABEL |
| 00000D | BRPLNLI | DS 0XL1 | |
| | 00001 BRPLNLM | EQU X'01' | |
| 00000D | BRPLFLBL | DS X TAPE FILE LABEL | |
| | * | | |
| | BRPLDAS | #FLAG X'80' | DATASET RESIDES ON DASD |
| 00000E | BRPLDASI | DS 0XL1 | |
| | 00080 BRPLDASM | EQU X'80' | |
| | BRPLMT | #FLAG X'40' | DATASET RESIDES ON MAG TAPE |
| 00000E | BRPLMTI | DS 0XL1 | |
| | 00040 BRPLMTM | EQU X'40' | |
| 00000E | BRPLUNIT | DS X UNIT BYTE | |
| | * | | |
| | BRPLBAS | #FLAG X'80' | ISSUE BASE STATS MSGS AT CLOSE |
| 00000F | BRPLBASI | DS 0XL1 | |
| | 00080 BRPLBASM | EQU X'80' | |
| | BRPLSUP | #FLAG X'40' | ISSUE SUPPLEMENTARY STATS MSGS AT CLOSE |
| 00000F | BRPLSUPI | DS 0XL1 | |
| | 00040 BRPLSUPM | EQU X'40' | |
| | BRPLSTY | #FLAG BRPLBASM+BRPLSUPM ISSUE BASE+SUPP MSG AT CLOSE | |
| 00000F | BRPLSTYI | DS 0XL1 | |

| | | | |
|--------|--------------|------------------|--|
| | 000C0 | BRPLSTYM EQU | BRPLBASM+BRPLSUPM |
| 00000F | BRPLOPT1 DS | X | OPTION FLAG BYTE 1 |
| | * | | |
| 000010 | BRPLFILE DS | CL8 | FILE IDENTIFICATION-DDNAME LINKNAME FILENAME |
| 000018 | BRPLSYS DS | CL6 | LOGICAL SYSTEM NUMBER (DOS/VSE ONLY) |
| 00001E | DS | XL2 | RESERVED |
| 000020 | BRPLBLEA DS | A | A(BLE) ASSOCIATED WITH THIS DATASET |
| 000024 | BRPLAREA DS | A | A(AREA) FOR RECORD |
| 000028 | BRPLRECL DS | F | DATASET LOGICAL RECORD LENGTH |
| 00002C | BRPLBLKS DS | F | DATASET BLOCKSIZE |
| 000030 | BRPL#INP DS | F | #INPUT OPERATIONS |
| 000034 | BRPL#OUT DS | F | #OUTPUT OPERATIONS |
| 000038 | BRPLMRBA DS | A | A(MRB) (MAPPING REQ) 0 (DIRECT-CALL REQ) |
| 00003C | BRPLCRBA DS | A | A(CHECKPOINT/RESTART BLOCK) |
| 000040 | BRPLDEVN DS | CL4 | DEVICE NAME |
| 000044 | BRPLULPN DS | CL8 | USER LABEL PROGRAM NAME (TAPE) |
| 00004C | DS | 5F | RESERVED |
| 00060 | BRPLLLEN EQU | ((*-BRPL+3)/4)*4 | LENGTH OF BRPL |

1.34 #BSBDS

```

COPY #BSBDS
*****
***          BSB: BATCH STATISTICS BLOCK
***          BSB IS A DSECT THAT DESCRIBES THE BATCH STATISTICS BLOCK ***
***          WRITTEN TO THE LOG WHICH CONTAINS THE STATISTICS
***          ACCUMULATIONS FOR ADS DIALOGS.
***          LENGTH OF HEADER
***          BATCH STATISTICS BLOCK BEGINNING
***          DIALOG VERSION NUMBER
***          ON IF AN @ACCEPT HAS BEEN DONE
***          ON IF INITIAL LINK LEVEL SET
***          ON IF INITIAL RBB ANALYSIS DONE
***          BSB FLAG BYTE
***          RESERVED
*****
**          THIS SECTION HOLDS CONTROL COMMAND COUNTERS FOR EXPLICITLY
**          CODED CONTROL COMMANDS
**          NUMBER OF INVOKE
**          LINK TO DIALOG
**          LINK TO PROGRAM
**          RETURN
**          RETURN CONTINUE
**          TRANSFER
**          LEAVE ADS
**          LEAVE APPLICATION
**          ABORT
**          WRITE TRANS
**          WRITE TRANS CONTINUE
**          WRITE TRANS RETURN
**          READ TRANS
**          CONTINUE
*****
**          THIS SECTION HOLDS CONTROL COMMAND COUNTERS FOR IMPLICIT ADS
**          GENERATED CONTROL COMMANDS
**          NUMBER OF IMPLICIT INVOKE
**          IMPLICIT LINK TO DIALOG
**          IMPLICIT LINK TO PROGRAM
**          IMPLICIT RETURN
**          IMPLICIT RETURN CONTINUE
**          IMPLICIT TRANSFER

```

| | | | | | |
|---|-----------|-----|-------|---|-------------------------------|
| 000090 | BSAILVD | DS | F | " | IMPLICIT LEAVE ADS |
| 000094 | BSAILVP | DS | F | " | IMPLICIT LEAVE APPLICATION |
| 000098 | BSAIAB | DS | F | " | IMPLICIT ABORT |
| 00009C | BSAIWTR | DS | F | " | IMPLICIT WRITE TRANS |
| 0000A0 | BSAIWTC | DS | F | " | IMPLICIT WRITE TRANS CONTINUE |
| 0000A4 | BSAIWTB | DS | F | " | IMPLICIT WRITE TRANS RETURN |
| 0000A8 | BSAIRDT | DS | F | " | IMPLICIT READ TRANS |
| 0000AC | BSAIC | DS | F | " | IMPLICIT CONTINUE |
| ***** | | | | | |
| ** | | | | | |
| ** THIS SECTION HOLDS GENERAL DIALOG EXECUTION STATISTICS | | | | | |
| ** | | | | | |
| ***** | | | | | |
| 0000B0 | BSAPMAP | DS | F | NUMBER OF PREMAP PROCESS EXECUTIONS | |
| 0000B4 | BSARESP | DS | F | " | RESPONSE PROCESS EXECUTIONS |
| 0000B8 | BSASTCL | DS | F | " | STATS ACCUMULATION CALLS |
| 0000BC | BSAWRPR | DS | F | " | WRITE TO PRINTER COMMANDS |
| 0000C0 | BSAWOPR | DS | F | " | WRITE TO OPERATOR |
| 0000C4 | BSAWLOG | DS | F | " | WRITE TO LOG |
| 0000C8 | BSAFDBSZ | DS | F | SIZE OF THE FDB | |
| 0000CC | BSAVDBSZ | DS | F | SIZE OF THE VDB | |
| 0000D0 | BSAHILEV | DS | H | HIGHEST LINK LEVEL AT WHICH DIALOG EXECUTED | |
| 0000D2 | BSALOLEV | DS | H | LOWEST LINK LEVEL AT WHICH DIALOG EXECUTED | |
| 0000D4 | BSASTIME | DS | F | SYSTEM CPU TIME | |
| 0000D8 | BSAUTIME | DS | F | USER CPU TIME | |
| ***** | | | | | |
| ** | | | | | |
| ** THIS SECTION HOLDS RBB USAGE STATISTICS | | | | | |
| ** | | | | | |
| ***** | | | | | |
| 0000DC | BSARBBSC | DS | F | NUMBER OF TIMES RBB'S PUT TO SCRATCH | |
| 0000E0 | BSAMSTG | DS | F | MOST RBB STORAGE USED (ALL DIALOGS) | |
| 0000E4 | BSAMSFRE | DS | F | ... RBB FREE SPACE WHEN MOST STORAGE USED | |
| 0000E8 | BSALSTG | DS | F | LEAST RBB STORAGE USED (ALL DIALOGS) | |
| 0000EC | BSALSFRE | DS | F | ... RBB FREE SPACE WHEN LEAST STORAGE USED | |
| 0000F0 | BSAMSBUF | DS | F | MOST RBB SPACE ACQUIRED FOR THIS DIALOG | |
| 0000F4 | BSALSBUF | DS | F | LEAST RBB SPACE ACQUIRED FOR THIS DIALOG | |
| 0000F8 | BSAHIRBB | DS | H | HIGHEST NUMBER OF RBB'S USED | |
| 0000FA | BSALORBB | DS | H | LOWEST NUMBER OF RBB'S USED | |
| PRINT GEN | | | | | |
| 000FC | BSBDSLLEN | EQU | *-BSB | LENGTH OF ADS STATS RECORD | |

1.35 #BSPDS

```
COPY #BSPDS
*****
***          BSP: ADS BATCH STATISTICS POOL
***          ***
***          ***
***          *** BSP IS A DSECT THAT DESCRIBES THE ADS STATISTICS POOL   ***
***          USED TO HOLD BATCH STATISTICS ACCUM BLOCKS (BSB) WHERE    ***
***          THE ADS RUNTIME SYSTEM COLLECTS DIALOG LEVEL STATISTICS. ***
***          THE BSP IS COMPRISED OF THE HEADER PORTION DESCRIBED    ***
***          HERE AND ONE BDS AND ONE BSB FOR EACH DIALOG WHICH HAS   ***
***          EXECUTED FOR THE CURRENT APPLICATION.                   ***
***          ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|-------------|--|
| 000000 | BSP | DSECT | 14:18:17 05/22/85 |
| | | PRINT NOGEN | |
| 000000 | BSPID | DS CL4 | 'BSP*' |
| 000004 | BSPNXTA | DS F | NEXT BSP ADDRESS |
| 000008 | BSPREVA | DS F | PREVIOUS BSP ADDRESS |
| 00000C | BSPINDX | DS H | INDEX OF THIS BSP |
| 00000E | BSPMAX | DS H | MAX NUMBER OF BDS/BSB'S PER BSP |
| 000010 | BSPNBSB | DS H | NUMBER OF BDS/BSB'S IN THIS BSP |
| 000012 | BSPASASZ | DS H | SIZE OF ONE BDS/BSB |
| 00014 | BSPDSLEN | EQU **-BSP | LENGTH OF BSP HEADER |
| | | | ***** |
| | | | *** |
| | | | *** FOLLOWING THE BSP HEADER ARE THE ACCUMULATION BLOCKS, *** |
| | | | *** ONE BDS AND ONE BSB FOR EACH DIALOG AND POSSIBLY ONE *** |
| | | | *** OF EACH FOR "ALL OTHER" DIALOGS, THAT IS, ALL DIALOGS *** |
| | | | *** FOR WHICH STATISTICS ARE NOT SPECIFICALLY REQUESTED. THE *** |
| | | | *** BDS COMES FIRST AND IS USED TO ACCUMULATE DB STATS. *** |
| | | | *** THEN COMES THE BSB CONTAINING STATS WHICH ARE *** |
| | | | *** PECULIAR TO ADS DIALOGS. THE BSB DSECT IS FOUND IN THE *** |
| | | | *** #BSBDS COPY MEMBER. *** |
| | | | *** |
| | | | ***** |
| | | | COPY #BDSDS |
| | | | ***** |
| | | | *** *** |
| | | | *** *** BDS: BATCH DATABASE STATISTICS BLOCK |
| | | | *** *** |
| | | | ***** |
| 000000 | BDS | DSECT | 09:33:41 04/15/86 |
| 000000 | BDSBID | DS CL4 | 'BDS*' |
| 000004 | BDSID | DS 0CL40 | BATCH DATABASE STATS ID |
| 000004 | BDSUSID | DS CL32 | USER ID |
| 000024 | BDSBATID | DS CL8 | "*BATCH**" |
| 00002C | BDSUID | DS CL8 | DIALOG NAME OR "\$ADS@AO" |
| 000034 | BDSUDAT | DS PL4 | DATE STATISTICS STARTED |
| 000038 | BDSUTIM | DS F | TIME STATISTICS STARTED |
| 00003C | BDSDB | DS 0F | START OF DB STATISTICS. |
| 00003C | BDSPAGRD | DS F | NUMBER OF PAGES READ. |
| 000040 | BDSPAGWR | DS F | NUMBER OF PAGES WRITTEN. |
| 000044 | BDSPAGRQ | DS F | NUMBER OF PAGES REQUESTED. |
| 000048 | BDSCALNO | DS F | NUMBER OF CALC RECS WITH NO OFLOW. |
| 00004C | BDSCALOF | DS F | NUMBER OF CALC RECS WITH OFLOW. |
| 000050 | BDSVIANO | DS F | NUMBER OF VIA RECS WITH NO OFLOW. |
| 000054 | BDSVIAOF | DS F | NUMBER OF VIA RECS WITH OFLOW. |
| 000058 | BDSRECRQ | DS F | NUMBER OF RECORDS REQUESTED. |
| 00005C | BDSRECCU | DS F | NUMBER OF RECS CURRENT OF R-U. |

| | | | | |
|--------|----------|--|-------|------------------------------|
| 000060 | | DS | F | NOT USED. |
| 000064 | BDSFRAGS | DS | F | NUMBER OF FRAGMENTS STORED. |
| 000068 | BDSRELO | DS | F | NUMBER OF RECORDS RELOCATED. |
| 00006C | BDSTLOCK | DS | F | TOTAL NUMBER OF LOCKS |
| 000070 | BDSSLOCK | DS | F | NUMBER OF SELECT LOCKS |
| 000074 | BDSULOCK | DS | F | NUMBER OF UPDATE LOCKS |
| | 0003C | BDSOSCG | EQU | *-BDSDB |
| 000078 | BDSESCG | DS | F | OFFSET TO GET SCRATCH |
| | 00040 | BDSOSCP | EQU | *-BDSDB |
| 00007C | BDSESCP | DS | F | NUMBER SCRATCH GETS |
| | 00044 | BDSOSCD | EQU | *-BDSDB |
| 000080 | BDSESCD | DS | F | OFFSET TO PUT SCRATCH |
| | 00048 | BDSOQUG | EQU | *-BDSDB |
| 000084 | BDSEQUG | DS | F | NUMBER SCRATCH PUTS |
| | 0004C | BDSOQUP | EQU | *-BDSDB |
| 000088 | BDSEQUP | DS | F | NUMBER SCRATCH DELETES |
| | 00050 | BDSOQUD | EQU | *-BDSDB |
| 00008C | BDSEQUD | DS | F | OFFSET TO DEL SCRATCH |
| | 00054 | BDSDSLEN | EQU | *-BDSDB |
| 000090 | BDSDECL | EQU | *-BDS | LENGTH OF DATA BLOCK |
| | | | | LENGTH OF DSECT |
| ***** | | | | |
| | *** | | | *** |
| | *** | FOLLOWING THE BDS IS THE BSB WHICH CONTAINS THE ADS | | |
| | *** | STATS FIELDS. THESE FIELDS IMMEDIATELY FOLLOW THE BDS. | | |
| | *** | SEE THE BSB DSECT IN THE #BSBDS COPY MEMBER. | | |
| | *** | | | *** |
| ***** | | | | |
| 000090 | BSABHERE | DS | OF | THE BSB STARTS HERE |

1.36 #CMEDS

```

COPY #CMEDS
*****
*** CME IS A DSECT THAT DEFINES AN ENTRY IN THE RESPONSE TABLE ***
*** OF THE FIXED DIALOG BLOCK (FDB) OF THE ADSO RUNTIME SYSTEM. ***
*****





| <u>Offset</u> | <u>Value</u>                                                         |                                     |
|---------------|----------------------------------------------------------------------|-------------------------------------|
|               |                                                                      | PRINT NOGEN                         |
| 000000        | CME DSECT                                                            | 03/09/88 11:58:37 05/20/92          |
|               | CMEBKPT #FLAG X'80'                                                  | ON = BREAKPOINT SET BY DEBUGGER     |
| 000000        | CMECLASS DS H                                                        | COMMAND ELEMENT MAJOR CLASS         |
| 000002        | CMEFUNC DS H                                                         | COMMAND ELEMENT FUNCTION            |
| 000004        | CMENXTA DS F                                                         | OFFSET OF NEXT CME                  |
| 000008        | CMEFLAG1 DS X                                                        | FIRST FLAG BYTE                     |
| 000009        | CMEFLAG2 DS X                                                        | SECOND FLAG BYTE                    |
| 00000A        | CMEFLAG3 DS X                                                        | THIRD FLAG BYTE                     |
|               | CMEAIDO #FLAG X'80'                                                  | SET IF CME IS ADSO RUNTIME BUILT    |
| 00000B        | CMEFLAG4 DS X                                                        | FOURTH FLAG BYTE                    |
| 00000C        | CMELEN EQU *--CME                                                    | LENGTH OF CME FOUNDATION            |
|               | CMEBODY DS 0H                                                        | START OF CLASS SPECIFIC DATA        |
|               | *                                                                    |                                     |
|               | *                                                                    |                                     |
|               | * IDMS/DML CALL (DATABASE) CME CLASS=1, ALL FUNCTIONS EXCEPT 99      |                                     |
|               | *                                                                    |                                     |
| 00000C        | ORG CMEFLAG1                                                         |                                     |
|               | *                                                                    |                                     |
|               | * THE FOLLOWING EQUATES ARE USED TO SPECIFY THE BIT SETTINGS FOR THE |                                     |
|               | * FIRST THREE FLAGS AS OPPOSED TO #FLAG MACROS WHICH BIND THE        |                                     |
|               | * BIT SETTINGS TO A SPECIFIC FLAG.                                   |                                     |
|               | *                                                                    |                                     |
| 00080         | CMEBRDEI EQU X'80'                                                   | ADE IS AN RDE INDEX                 |
| 00040         | CMEBSSAO EQU X'40'                                                   | ADE IS AN SSAN TABLE OFFSET         |
| 00040         | CMEBPRFX EQU X'40'                                                   | MSG PREFIX SPECIFIED *MCM86178*     |
| 00020         | CMEBFWD EQU X'20'                                                    | ADE IS AN EMBEDDED FULLWORD         |
| 00010         | CMEBFM2 EQU X'10'                                                    | ADE IS LITPOOL OFFSET & LENGTH      |
| 00010         | CMEBPFBA EQU X'10'                                                   | OFF/ON - MSG PREFIX COMES*MCM86178* |
|               | *                                                                    | BEFORE/AFTER PARMS *MCM86178*       |
| 00008         | CMEBFM3 EQU X'08'                                                    | ADE IS FORMAT 3 (4 HALFWORDS)       |
| 00004         | CMEBXDES EQU X'04'                                                   | ADE IS FOLLOWED BY AN XDE STACK TO  |
|               | *                                                                    | OBTAIN A SUBSCRIPTED FIELD          |
| 00002         | CMEBDBK EQU X'02'                                                    | PARM IS A DATABASE KEY              |
| 00002         | CMEBPFM3 EQU X'02'                                                   | FMT 3 ADE USED FOR PREFIX*MCM86178* |
| 00001         | CMEBSK EQU X'01'                                                     | PARM IS A SORT KEY                  |
| 00001         | CMEBPXDE EQU X'01'                                                   | PREFIX ADE FOLL'D BY XDES*MCM86178* |
| 000008        | CMEBFLG1 DS X                                                        | FIRST DML COMMAND FLAG              |
| 000009        | CMEBFLG2 DS X                                                        | SECOND DML COMMAND FLAG             |
| 00000A        | CMEBFLG3 DS X                                                        | THIRD DML COMMAND FLAG              |
|               | CMEBPOB #FLAG X'40'                                                  | PIGGYBACK OBTAIN FUNCTION           |
|               | CMEBPKP #FLAG X'20'                                                  | PIGGYBACK KEEP FUNCTION             |
|               | CMEBPKE #FLAG X'10'                                                  | PIGGYBACK KEEP EXCLUSIVE FUNCTION   |
|               | CMEBRTV #FLAG X'08'                                                  | RETRIEVAL CME                       |
|               | CMEBEC #FLAG X'04'                                                   | ON IF ERROR CODE LIST EXISTS        |
| 00000B        | CMEBFLG4 DS X                                                        | FOURTH DML COMMAND FLAG             |
| 00000C        | CMEBSZ EQU *--CME                                                    | SIZE OF BASIC DML COMMAND CME       |
| 00000C        | CMEBADE DS 0F                                                        | START OF ARGUMENT DESCRIPTION       |
|               | *                                                                    | ELEMENT(S) (SEE ADE DSECT)          |
|               | *                                                                    |                                     |
|               | *                                                                    |                                     |
|               | * IDMS/DML CALL (LOGICAL RECORD) CME CLASS=1, FUNCTION=99            |                                     |
|               | *                                                                    |                                     |
| 00000C        | ORG CMEFLAG1                                                         |                                     |


```

```

CMECWHR #FLAG X'80' WHERE CLAUSE (XDE STACK) INCLUDED
CMECOLR #FLAG X'40' "OF LR" XDE IN THE STACK
000008 CMECFLG1 DS X FIRST LR COMMAND FLAG
000009 CMECFLG2 DS X SECOND LR COMMAND FLAG
00000A CMECFLG3 DS X THIRD LR COMMAND FLAG
** FOLLOWING FLAG ADDRESSES THE SAME BIT AS FOUND IN NON-LR DML CME'S
CMECRTV #FLAG X'08' RETRIEVAL CME
00000B CMECFLG4 DS X FOURTH LR COMMAND FLAG
00000C CMECVBLN DS H LENGTH OF DML VERB LITERAL
00000E CMECVBOF DS H LITPOOL OFFSET TO DML VERB LITERAL
000010 CMECLRI DS H RDE INDEX OF LOGICAL RECORD
000012 DS H RESERVED
00014 CMECSZ EQU *-CME SIZE OF BASIC LR COMMAND CME
000014 CMECXSTK DS OF START OF XDE STACK IF PRESENT
*
*
* ABORT CME CLASS=2, FUNCTION=0
*
000014 ORG CMEFLAG1
CMEABTO #FLAG X'80' ABORT ADSO
CMEABTA #FLAG X'40' ABORT APPLICATION
CMEABTS #FLAG X'10' SNAP RESOURCES ON ABORT
000008 CMEAFLG1 DS X FIRST ABORT FLAG BYTE
* CMEDISP ('80') AND CMEDISL ('20') FROM DISPLAY ARE USED FOR ABORT
* ALSO CMEDSMS ('40')
000009 CMEAFLG2 DS X SECOND ABORT FLAG BYTE
* CMEBPRFX('40'), CMEBPFBA('10'), CMEBPFM3('02') AND *MCM86178*
* CMEBPXDE('01') USED IN CMEAFLG3 FOR MSG PREFIX INFO *MCM86178*
00000A CMEAFLG3 DS X THIRD ABORT FLAG BYTE
* CMEBFM3 ('08') AND CMEBXDES ('04') ADE BITS ARE SET IN CMEAFLG4
00000B CMEAFLG4 DS X FOURTH ABORT FLAG BYTE
00000C CMEABSZ EQU *-CME SIZE OF BASIC ABORT CME
00000C CMEAERRC DS F ERROR CODE OF ABORT MESSAGE
000004 CMEABMSZ EQU *-CMEAERRC SIZE OF BASIC MESSAGE DESCRIPTOR
000010 CMEABTSZ EQU *-CME SIZE OF ABORT CME
000010 CMEABLIT DS H START OF INLINE LITERAL ADS370/75
000012 CMEABRSV DS H **** RESERVED **** ADS370/75
00014 CMEABLSZ EQU *-CME SZ OF AB CME WITH LIT ADS370/75
*
*
* INVOKE CME CLASS=2, FUNCTION=1
*
000014 ORG CMEFLAG1 *JMA85058*
CMELNNS #FLAG X'80' NOSAVE/NORESTORE CURRENCY*JMA85058*
000008 CMELNFL1 DS X FIRST INVOKE FLAG *JMA85058*
000009 CMELNFL2 DS X SECOND INVOKE FLAG *JMA85058*
00000A CMELNFL3 DS X THIRD INVOKE FLAG *JMA85058*
00000B CMELNFL4 DS X FOURTH INVOKE FLAG *JMA85058*
00000C CMELNKSZ EQU *-CME SIZE OF INVOKE COMMAND ENTRY
*
*
* TRANSFER CME CLASS=2, FUNCTION=2
*
00000C ORG CMEFLAG1 *JMA85058*
CMETRNF #FLAG X'80' NOFINISH SPECIFIED *JMA85058*
000008 CMETRFL1 DS X FIRST TRANSFER FLAG *JMA85058*
000009 CMETRFL2 DS X SECOND TRANSFER FLAG *JMA85058*
00000A CMETRFL3 DS X THIRD TRANSFER FLAG *JMA85058*
00000B CMETRFL4 DS X FOURTH TRANSFER FLAG *JMA85058*
00000C CMETRSZ EQU *-CME SIZE OF TRANSFER CME
*
*
* RETURN CME CLASS=2, FUNCTION=3
*
00000C ORG CMEFLAG1
CMERETC #FLAG X'80' RET CONTINUE (DO PREMAP PROCESS)

```

```

000008      CMERETI  #FLAG X'40'          RETURN CLEAR (INITIALIZE)
            CMERFLG1 DS   X               FIRST RETURN FLAG BYTE
            CMERETA #FLAG X'80'          RETURN ALL TO TOP DIALOG
            CMERETN #FLAG X'40'          RETURN DIALOG NAME GIVEN
000009      CMERFLG2 DS   X               SECOND RETURN FLAG BYTE
00000A      CMERFLG3 DS   X               THIRD RETURN FLAG BYTE
00000B      CMERFLG4 DS   X               FOURTH RETURN FLAG BYTE
00000C      CMERETSZ EQU  *--CME        SIZE OF RETURN CME
*
*
*  DISPLAY CME  CLASS=2, FUNCTION=4
*
00000C      ORG    CMEFLAG1
            CMEDISC #FLAG X'80'          DISP CONTINUE (DO PREMAP PROCESS)
            CMEDISM #FLAG X'40'          DISPLAY WITH MESSAGE
            CMEDSNS #FLAG X'20'          DISPLAY NOSAVE
            CMEDISE #FLAG X'10'          DISPLAY ERASE      *RQE84227*
000008      CMEDSFL1 DS   X               FIRST DISPLAY FLAG BYTE
** THE FOLLOWING 3 FLAG BIT SETTINGS ARE ALSO USED IN THE *JEB84057*
** ABORT, PUT DETAIL, WRITE TRAN AND CONTINUE CME'S      *DKJ85128*
            CMEDISP #FLAG X'80'          REPLACEMENT PARMS
            CMEDSMS #FLAG X'40'          MSG IS SYSTEM GENERATED *LCB85176*
            CMEDISL #FLAG X'20'          LITERAL USED
000009      CMEDSFL2 DS   X               SECOND DISPLAY FLAG BYTE
* CMEBPRFX('40'), CMEBFM3('10') AND      *MCM86178*
* CMEBPXDE('01') USED IN CMEAFLG3 FOR MSG PREFIX INFO *MCM86178*
00000A      CMEDSFL3 DS   X               THIRD DISPLAY FLAG BYTE
00000B      CMEDSFL4 DS   X               FOURTH DISPLAY FLAG BYTE
00000C      CMEDSSZ EQU  *--CME        SIZE OF BASIC DISPLAY CME
00000C      CMEDSLSZ DS   H               SIZE OF LITERAL FOR MESSAGE
00000E      CMEDSLOF DS   H               OFFSET TO LITERAL IN LITPOOL
000010      ORG    CMEDSLSZ
00000C      CMEDSLKY DS   F               CODE FOR PRESTORED MESSAGE
000004      CMEDSMSZ EQU  *--CMEDSLSZ  SIZE OF BASIC MSG DESCRIPTOR
000010      CMEDSLIT DS   0H              START OF INLINE LITERAL
000010      CMEMSPNO DS   F               NUM OF MSG REPLACEMENT PARMS
000014      CMEDSRPL DS   0F              START OF REP PARM LIST
*****
* IF REPLACEMENT PARMS ARE SPECIFIED, THERE WILL BE ONE REPLACEMENT *
* PARAMETER ELEMENT (RPE) HERE FOR EACH PARM (SEE RPE DSECT).      *
*****
*
*
*  LEAVE CME  CLASS=2, FUNCTION=6
*
000014      ORG    CMEFLAG1
            CMELVEO #FLAG X'80'         LEAVE ADSO
            CMELVEA #FLAG X'40'         LEAVE APPLICATION
            CMELVER #FLAG X'20'         RESUME ABEND (DEBUGGING DIALOG)
            CMECOND #FLAG X'10'         CONDITION CODE (BATCH)
000008      CMELVFL1 DS   X               FIRST LEAVE FLAG BYTE
            CMELVEN #FLAG X'80'         LEAVE APPL, NEXT DIALOG NAMED
            CMELVEX #FLAG X'40'         NEXT TASK NAMED      *JEB85144*
000009      CMELVFL2 DS   X               SECOND LEAVE FLAG BYTE
00000A      CMELVFL3 DS   X               THIRD LEAVE FLAG BYTE
00000B      CMELVFL4 DS   X               FOURTH LEAVE FLAG BYTE
00000C      CMELVESZ EQU  *--CME        SIZE OF LEAVE CME
*
*
*  LINK CME  CLASS=2, FUNCTION=7
*
00000C      ORG    CMEFLAG1
            CMEINUP #FLAG X'80'         LINK TO USER PROGRAM
            CMEINVD #FLAG X'40'         LINK TO DIALOG
            CMEINNS #FLAG X'20'         NOSAVE/NORESTORE CURRENCIES
000008      CMEINVF1 DS   X               FIRST LINK FLAG

```

| | | |
|--------|---|--|
| | CMEINUR #FLAG X'80' | LINK TO USER WITH USING RECS |
| | CMEINUS #FLAG X'40' | LINK TO DIALOG WITH SUBSCHEMA CTL |
| | CMEINUM #FLAG X'20' | LINK TO DIALOG WITH MAP CTRL |
| | CMEINUL #FLAG X'10' | LNK TO USR WITH USING LITR*LMA88032* |
| 000009 | CMEINVF2 DS X | SECOND LINK FLAG |
| 00000A | CMEINVF3 DS X | THIRD LINK FLAG |
| 00000B | CMEINVF4 DS X | FOURTH LINK FLAG |
| 00000C | CMEINVSZ EQU *-CME | SIZE OF LINK TO DIALOG/USER PR CME *JMA85058* |
| | * | *JMA85058* |
| | * | *JMA85058* |
| | * WRITE TRANSACTION CME | CLASS=2, FUNCTION=8 |
| | * | *JMA85058* |
| 00000C | ORG CMEFLAG1 | *JMA85058* |
| | CMEWTC #FLAG X'80' | WRITE TRANSACTION CONTIN *JMA85058* |
| | CMEWTM #FLAG X'40' | WRITE WITH MESSAGE *JMA85058* |
| | CMEWTR #FLAG X'20' | READ SPECIFIED *JMA85058* |
| | CMEWTRE #FLAG X'10' | RETURN SPECIFIED *DKJ85128* |
| | CMEWTRT #FLAG X'08' | RETURN TO TOP SPECIFIED *DKJ85128* |
| | CMEWTRD #FLAG X'04' | RETURN TO DIALOG *DKJ85128* |
| | CMEWTRC #FLAG X'02' | RETURN CLEAR *DKJ85128* |
| | CMEWTRN #FLAG X'01' | RETURN CONTINUE *DKJ85128* |
| 000008 | CMEWTF1 DS X | FIRST WRITE TRAN FLAG *JMA85058* |
| | * CMEDISP ('80') AND CMEDISL ('20') FROM DISPLAY ARE USED FOR FLAGS | |
| | * ALSO CMEDSMS ('40') | |
| | CMEWTRS #FLAG X'10' | WRITE TRAN SUSPENSE *RQE86010* |
| 000009 | CMEWTF12 DS X | SECOND WRITE TRAN FLAG *JMA85058* |
| | * CMEBPRFX('40'), CMEBPFBA('10'), CMEBPFM3('02') AND | *MCM86178* |
| | * CMEBPXDE('01') USED IN CMEAFLG3 FOR MSG PREFIX INFO | *MCM86178* |
| 00000A | CMEWTF13 DS X | THIRD WRITE TRAN FLAG *JMA85058* |
| 00000B | CMEWTF14 DS X | FOURTH WRITE TRAN FLAG *JMA85058* |
| 00000C | CMEWTSZ EQU *-CME | SIZE OF BASIC WRITE CME *JMA85058* |
| 00000C | CMEWTLSZ DS H | LIT SIZE FOR MESSAGE *JMA85058* |
| 00000E | CMEWTLOF DS H | LIT OFFSET IN LIT POOL *JMA85058* |
| 000010 | ORG CMEWTLSZ | *JMA85058* |
| 00000C | CMEWTLKY DS F | PRESTORED MESSAGE CODE *JMA85058* |
| 000004 | CMEWTMSZ EQU *-CMEWTLSZ | BASIC MSG DESCRIPT SIZE *JMA85058* |
| 000010 | CMEWTLIT DS OH | START OF INLINE LITERAL *JMA85058* |
| 000010 | CMEWTMPN DS F | NUM OF MSG REPLACEMENT PARM*JMA85058* |
| 000014 | CMEWTRPL DS OF | START OF REP PARM LIST *JMA85058* |
| | ***** | ***** |
| | * IF REPLACEMENT PARMS ARE SPECIFIED, THERE WILL BE ONE REPLACEMENT * | |
| | * PARAMETER ELEMENT (RPE) HERE FOR EACH PARM (SEE RPE DSECT). * | |
| | ***** | ***** |
| | * | *JMA85058* |
| | * | *JMA85058* |
| | * READ TRANSACTION CME | CLASS=2, FUNCTION=9 |
| | * | *JMA85058* |
| | * | *JMA85058* |
| 000014 | ORG CMEFLAG1 | *DKJ85149* |
| | CMERTIO #FLAG X'80' | INPUT-OUTPUT *DKJ85149* |
| 000008 | CMERTFL1 DS X | FIRST READ TRAN FLAG *DKJ85149* |
| 000009 | CMERTFL2 DS X | SECOND READ TRAN FLAG *DKJ85149* |
| 00000A | CMERTFL3 DS X | THIRD READ TRAN FLAG *DKJ85149* |
| 00000B | CMERTFL4 DS X | FOURTH READ TRAN FLAG *DKJ85149* |
| 00000C | CMERTSZ EQU *-CME | SIZE OF READ TRAN CME *JMA85058* |
| | * | *JMA85058* |
| | * | *JMA85058* |
| | * CONTINUE CME CLASS=2, FUNCTION=10 | *JMA85058* |
| | * | *JMA85058* |
| 00000C | ORG CMEFLAG1 | *JMA85058* |
| | CMECNM #FLAG X'80' | CONTINUE WITH MESSAGE *JMA85058* |
| 000008 | CMECNFL1 DS X | FIRST CONTINUE FLAG *JMA85058* |
| | * CMEDISP ('80') AND CMEDISL ('20') FROM DISPLAY ARE USED FOR FLAGS | |
| | * ALSO CMEDSMS ('40') | |
| 000009 | CMECNFL2 DS X | SECOND CONTINUE FLAG *JMA85058* |
| | * CMEBPRFX('40'), CMEBPFBA('10'), CMEBPFM3('02') AND | *MCM86178* |

```

        * CMEBPXDE('01') USED IN CMEAFLG3 FOR MSG PREFIX INFO      *MCM86178*
00000A    CMECNFL3 DS   X          THIRD CONTINUE FLAG      *JMA85058*
00000B    CMECNFL4 DS   X          FOURTH CONTINUE FLAG     *JMA85058*
00000C    CMECNSZ EQU   *-CME      SIZE OF BASIC CONTIN CME *JMA85058*
00000C    CMECNLsz DS   H          LIT SIZE FOR MESSAGE    *JMA85058*
00000E    CMECNLOF DS   H          LIT OFFSET IN LIT POOL *JMA85058*
000010    ORG    CMECNLsz
00000C    CMECNLKY DS   F          PRESTORED MESSAGE CODE *JMA85058*
000004    CMECNMSz EQU   *-CMECNLsz BASIC MSG DESCRIPT SIZE *JMA85058*
000010    CMECNLIT DS   0H         START OF INLINE LITERAL *JMA85058*
000010    CMECNMPN DS   F          NUM OF MSG REPLCMENT PARM*JMA85058*
000014    CMECNRPL DS   0F         START OF REP PARM LIST *JMA85058*
*****
* IF REPLACEMENT PARMS ARE SPECIFIED, THERE WILL BE ONE REPLACEMENT **
* PARAMETER ELEMENT (RPE) HERE FOR EACH PARM (SEE RPE DSECT)      *
*****
*                                                               **SQL**
*                                                               **SQL**
* @SQLCALL  CME      CLASS=2, FUNCTION=11                      **SQL**
*                                                               **SQL**
000014    ORG    CMEFLAG1
000008    CMESQLF1 DS   X          FIRST @SQLCALL FLAG       **SQL**
000009    CMESQLF2 DS   X          SECOND @SQLCALL FLAG      **SQL**
00000A    CMESQLF3 DS   X          THIRD @SQLCALL FLAG      **SQL**
00000B    CMESQLF4 DS   X          FOURTH @SQLCALL FLAG     **SQL**
00000C    CMESQLSz EQU   *-CME      SIZE OF BASIC @SQLCALL CME **SQL**
00000C    CMESQRPB DS   H          SQLRPB RECORD INDEX     **SQL**
00000E    CMESQSSI DS   H          SQLSSI RECORD INDEX     **SQL**
000010    CMESQCA DS   H          SQLCA RECORD INDEX      **SQL**
000012    CMESQCIB DS   H          SQLCIB RECORD INDEX     **SQL**
000014    CMESQPIB DS   H          SQLPIB RECORD INDEX     **SQL**
000016    CMESQPBF DS   H          SQLPBF RECORD INDEX     **SQL**
000018    CMESQTBF DS   H          SQLTBF RECORD INDEX     **SQL**
00001A    CMESQDAI DS   H          SQLDA INPUT RECORD INDEX **SQL**
00001C    CMESQDAO DS   H          SQLDA OUTPUT RECORD INDEX **SQL**
00001E    CMESQEXS DS   H          SQLEXS RECORD INDEX     **SQL**
000020    CMESQFMT DS   H          SQLFMT RECORD INDEX     **SQL**
000022    DS      H          FILLER FOR FULLWD ALIGNMNT **SQL**
00018    CMESQISz EQU   *-CMESQRPB SIZE OF INDEX LIST      **SQL**
*                                                               **SQL**
*                                                               **SQL**
*****
*                                                               *
*   ASSIGNMENT CME  CLASS=3, FUNCTION=1
*   "        CME   CLASS=3, FUNCTION=21 (ADD)
*   "        CME   CLASS=3, FUNCTION=22 (SUBTRACT)
*   "        CME   CLASS=3, FUNCTION=23 (MULTIPLY)
*   "        CME   CLASS=3, FUNCTION=24 (DIVIDE)
*   "        CME   CLASS=3, FUNCTION=40 (MOVE)
*   "        CME   CLASS=3, FUNCTION=41 (COMPUTE)
*   CONDITIONAL CME CLASS=3, FUNCTION=2
*   WHILE/REPEAT CME CLASS=3, FUNCTION=3
*   *
*   *
000024    ORG    CMEBODY
00000C    CMEXDEHA DS   F          OFFSET TO XDE STACK HEADER
00010    CMEA SGsz EQU   *-CME      SIZE OF ASG CME HEADER
000010    CMETRUEA DS   F          OFFSET OF TRUE CME START
000014    CMEFLSEA DS   F          OFFSET OF FALSE CME START
00018    CMECNDsz EQU   *-CME      SIZE CONDITIONAL/WHILE-REPEAT HDR
*
*****
*                                                               *
*   THE XDE STACK HEADER FALLS IMMEDIATELY AFTER THE CME HEADER;  *
*   . AFTER CMEXDEHA IN AN ASSIGNMENT CME.                         *
*   . AFTER CMEFLSEA IN A CONDITIONAL OR WHILE/REPEAT CME.        *

```

```

*
* THE XDE STACK HEADER CONSISTS OF FOUR HALWORDS; *
* 1ST HALWORD = NUMBER OF XDE'S AND DXB'S IN THE XDE STACK. *
* 2ND HALWORD = NUMBER OF IRA BYTES NEEDED TO EVALUATE. *
* 3RD HALWORD = XDE STACK SIZE NOT INCLUDING THE HEADER ITSELF. *
* 4TH HALWORD = UNUSED *
*
* IF HIGH ORDER (LEFTMOST) BIT OF 1ST HALWORD IS ON, THE *
* XDE STACK CONTAINS ONE OR MORE DXB'S; IF OFF, STACK *
* CONTAINS XDE'S ONLY. *
*
* THE XDE'S THEMSELVES IMMEDIATELY FOLLOW THE XDE STACK HEADER. *
*
*****
*
* INTERNAL BRANCH CME CLASS=3, FUNCTION=4
*
000018      ORG    CMEFLAG1
              CMEBRCX #FLAG X'80'           BRANCH IS FOR EXIT
              CMEBRCN #FLAG X'40'           BRANCH IS FOR NEXT-COMMAND
000008      ORG    CMEBODY
00000C      CMEBRCHA DS   F           OFFSET OF CME TO BRANCH TO
00010       CMEBRCSZ EQU   *-CME        SIZE OF BRANCH CME HEADER
*
*
* SUBROUTINE CALL CME CLASS=3, FUNCTION=5
*
000010      ORG    CMEBODY
00000C      CMESBRA DS   F           OFFSET OF FIRST CME IN SUBROUTINE
000010      CMESBRNM DS   CL8          SUBROUTINE LABEL NAME
00018       CMESBRSZ EQU   *-CME        SIZE OF SUBROUTINE CME HDR
*
*
* ON PATH STATUS CME CLASS=3, FUNCTION=7
*
000018      ORG    CMEBODY
00000C      CMEOPSLN DS  H           LENGTH OF PATH STATUS LITERAL
00000E      CMEOPSOF DS  H           LITPOOL OFFSET TO PATH STATUS LIT.
000010      CMEOTRUE DS  F           OFFSET TO TRUE CME START
000014      CMEOFLSE DS  F           OFFSET TO FALSE CME START
00018       CMEONPSZ EQU   *-CME        SIZE OF ON PATH STATUS CME
*
*
* MODIFY MAP CME CLASS=4, FUNCTION=1
*
000018      ORG    CMEFLAG1
              CMEMFRL #FLAG X'80'           CURSOR ROW IS LITERAL
              CMEMFCL #FLAG X'40'           CURSOR COLUMN IS LITERAL
              CMEMFRX #FLAG X'20'           CURSOR ROW HAS XDE STACK
              CMEMFCX #FLAG X'10'           CURSOR COLUMN HAS XDE STACK
              CMEMFPL #FLAG X'08'           PAD CHARACTER IS LITERAL
              CMEMFPX #FLAG X'04'           PAD CHARACTER HAS XDE STACK
000008      CMEMFLG1 DS   X           1ST MODIFY MAP FLAG BYTE
              CMEMLST #FLAG X'40'           MAPFIELD LIST
              CMEMFDM #FLAG X'20'           DFLD MRE INDEX LIST
              CMEMFDP #FLAG X'10'           DFLD LITPOOL OFFSET
000009      CMEMFLG2 DS   X           2ND MODIFY MAP FLAG BYTE
              CMEMFCM #FLAG X'20'           CURSOR MRE INDEX LIST
              CMEMFCP #FLAG X'10'           CURSOR LITPOOL OFFSET
00000A      CMEMFLG3 DS   X           3RD MODIFY MAP FLAG BYTE
00000B      CMEMFLG4 DS   X           4TH MODIFY MAP FLAG BYTE
00000C      CMEMATT C DS   F           ATTRIBUTE CHANGE BYTES
000010      ORG    CMEMATT C
              CMEMAWH #FLAG X'01'           WHITE
00000C      CMEMATT1 DS   X           1ST ATTRIBUTE CHANGE BYTE

```

| | | | |
|--------|----------|-------------|-------------------------------------|
| | CMEMAYL | #FLAG X'80' | YELLOW |
| | CMEMATQ | #FLAG X'40' | TURQUOISE |
| | CMEMAGR | #FLAG X'20' | GREEN |
| | CMEMAPK | #FLAG X'10' | PINK |
| | CMEMARD | #FLAG X'08' | RED |
| | CMEMABL | #FLAG X'04' | BLUE |
| | CMEMANC | #FLAG X'02' | NOCOLOR |
| | CMEMANS | #FLAG X'01' | NOUNDERSCORE |
| 00000D | CMEMATT2 | DS X | 2ND ATTR CHANGE BYTE |
| | CMEMAMU | #FLAG X'80' | UNDERSCORE |
| | CMEMANV | #FLAG X'40' | NORMAL-VIDEO |
| | CMEMARV | #FLAG X'20' | REVERSE-VIDEO |
| | CMEMANB | #FLAG X'10' | NOBLINK |
| | CMEMABK | #FLAG X'08' | BLINK |
| | CMEMANM | #FLAG X'04' | NOMDT |
| | CMEMAMD | #FLAG X'02' | MDT |
| | CMEMADE | #FLAG X'01' | DETECT |
| 00000E | CMEMATT3 | DS X | 3RD ATTRIBUTE CHANGE BYTE |
| | CMEMABR | #FLAG X'80' | BRIGHT |
| | CMEMADK | #FLAG X'40' | DARK |
| | CMEMADS | #FLAG X'20' | DISPLAY |
| | CMEMAUP | #FLAG X'10' | UNPROTECTED |
| | CMEMAPR | #FLAG X'08' | PROTECTED |
| | CMEMANU | #FLAG X'04' | NUMERIC |
| | CMEMAAL | #FLAG X'02' | ALPHANUMERIC |
| | CMEMASK | #FLAG X'01' | SKIP |
| 00000F | CMEMATT4 | DS X | 4TH ATTRIBUTE CHANGE BYTE |
| 000010 | CMEMGFLC | DS H | GENERAL FLAG BYTES |
| 000012 | ORG | CMEMGFLC | |
| | CMEMGMF | #FLAG X'01' | FOR DFLD OR (...) |
| | CMEMGEM | #FLAG X'02' | FOR ALL BUT DFLD OR (...) |
| | CMEMGEC | #FLAG X'04' | FOR ALL BUT CURRENT |
| 000010 | CMEMGFL1 | DS X | 1ST GENERAL FLAG BYTE |
| | CMEMGTC | #FLAG X'01' | ON FOR TEMPORARY MAP CHANGE |
| | CMEMGCR | #FLAG X'02' | ON FOR ROW/COL CURSOR POSITION |
| | CMEMGWC | #FLAG X'04' | ON IF WCC CHANGED |
| | CMEMGCF | #FLAG X'08' | ON FOR MAP FLD CURSOR POSITION |
| | CMEMGC | #FLAG X'10' | FOR CURRENT |
| | CMEMGAF | #FLAG X'20' | FOR ALL FIELDS |
| | CMEMGAC | #FLAG X'40' | FOR ALL CORRECT FIELDS |
| | CMEMGAE | #FLAG X'80' | FOR ALL ERROR FIELDS |
| 000011 | CMEMGFL2 | DS X | 2ND GENERAL FLAG BYTE |
| 000012 | CMEMOFLC | DS H | OUTPUT FLAG BYTES |
| 000014 | ORG | CMEMOFLC | |
| 000012 | CMEMOFL1 | DS X | 1ST OUTPUT FLAG BYTE |
| | CMEMOOA | #FLAG X'20' | OUTPUT DATA IS ATTRIBUTE *LCB84291* |
| | CMEMOOE | #FLAG X'10' | OUTPUT DATA IS ERASE |
| | CMEMOON | #FLAG X'08' | OUTPUT DATA IS NO |
| | CMEMOOY | #FLAG X'04' | OUTPUT DATA IS YES |
| | CMEMONB | #FLAG X'02' | NOBACKSCAN |
| | CMEMOB | #FLAG X'01' | BACKSCAN |
| 000013 | CMEMOFL2 | DS X | 2ND OUTPUT FLAG BYTE |
| 000014 | CMEMWCCC | DS H | WCC FLAG BYTES |
| 000016 | ORG | CMEMWCCC | |
| | CMEMW80 | #FLAG X'08' | 80CR |
| | CMEMW64 | #FLAG X'04' | 64CR |
| | CMEMW40 | #FLAG X'02' | 40CR |
| | CMEMWNC | #FLAG X'01' | NLCR |
| 000014 | CMEMWCC1 | DS X | 1ST WCC CHANGE BYTE |
| | CMEMWNP | #FLAG X'80' | NOPRT |
| | CMEMWSP | #FLAG X'40' | STARTPRT |
| | CMEMWNA | #FLAG X'20' | NOALARM |
| | CMEMWA | #FLAG X'10' | ALARM |
| | CMEMWNK | #FLAG X'08' | NOKBD |
| | CMEMWRK | #FLAG X'04' | RESETKBD |
| | CMEMWNM | #FLAG X'02' | NOMDT |

| | | |
|--------|--|-------------------------------------|
| | CMEMWRM #FLAG X'01' | RESETMDT |
| 000015 | CMEMWCC2 DS X | 2ND WCC CHANGE BYTE |
| 000016 | CMEMIFLC DS H | INPUT FLAG BYTES |
| 000018 | ORG CMEMIFLC | |
| | CMEMIMA #FLAG X'20' | ERROR MESSAGE IS ACTIVE *MCM86169* |
| | CMEMIMS #FLAG X'10' | ERROR MESSAGE IS SUPPRESS*MCM86169* |
| | CMEMIEC #FLAG X'02' | EDIT IS CORRECT |
| | CMEMIEE #FLAG X'01' | EDIT IS ERROR |
| 000016 | CMEMIFL1 DS X | 1ST INPUT FLAG BYTE |
| | CMEMIOP #FLAG X'80' | OPTIONAL |
| | CMEMIRQ #FLAG X'40' | REQUIRED |
| | CMEMILJ #FLAG X'20' | LEFT JUSTIFY |
| | CMEMIRJ #FLAG X'10' | RIGHT JUSTIFY |
| | CMEMINP #FLAG X'08' | NOPAD |
| | CMEMIP #FLAG X'04' | PAD |
| | CMEMIIN #FLAG X'02' | INPUT DATA IS NO |
| | CMEMIY #FLAG X'01' | INPUT DATA IS YES |
| 000017 | CMEMIFL2 DS X | 2ND INPUT FLAG BYTE |
| 000018 | CMEMCRA0 DS F | CURSOR ROW ADE OFFSET |
| 00001C | ORG CMEMCRA0 | |
| 000018 | CMEMCROW DS H | CURSOR ROW |
| 00001A | ORG CMEMCRA0 | |
| 000018 | CMEMCMIL DS H | CURSOR ROW MRE INDEX |
| 00001A | ORG CMEMCRA0+4 | |
| 00001C | CMEMCCAO DS F | CURSOR COLUMN ADE OFFSET |
| 000020 | ORG CMEMCCAO | |
| 00001C | CMEMCCOL DS H | CURSOR COLUMN |
| 00001E | ORG CMEMCCAO+4 | |
| 000020 | CMEMPDAO DS F | PAD CHARACTER ADE OFFSET |
| 000024 | ORG CMEMPDAO | |
| 000020 | CMEMPADC DS X | PAD CHARACTER |
| 000021 | ORG CMEMPDAO+4 | |
| 000024 | CMEMDMIL DS H | MAP FIELD MRE INDEX |
| 000026 | CMEMFIL1 DS H | EXTEND CME TO FULLWORD MULTIPLE |
| 000028 | CMEMMLEN EQU *-CME | SIZE OF MODIFY MAP CME HDR |
| 000028 | CMEMADE DS 0F | START OF 1ST ADE (IF ANY) |
| | * | *JEB84057* |
| | * | *JEB84057* |
| | * PUT DETAIL CME CLASS=4, FUNCTION=2 | *JEB84057* |
| | * | *JEB84057* |
| 000028 | ORG CMEFLAG1 | |
| | CMEPDCR #FLAG X'80' | PUT CURRENT DETAIL |
| | CMEPDMS #FLAG X'40' | MESSAGE INCLUDED |
| 000008 | CMEPDFL1 DS X | FIRST PUT DETAIL FLAG |
| | * CMEDISP ('80') AND CMEDISL ('20') FROM DISPLAY ARE USED | *JEB84057* |
| | * WHEN PUT WITH MESSAGE SPECIFIED | *JEB84057* |
| | * ALSO CMEDSMS ('40') | |
| 000009 | CMEPDFL2 DS X | SECOND PUT DETAIL FLAG |
| | * CMEBIFWD ('20'), CMEBFM3 ('08') AND CMEBXDES ('04') ADE | *JEB84057* |
| | * BITS ARE SET IN CMEPDFL3 WHEN PUT WITH KEY IS SPECIFIED | *JEB84057* |
| | * CMEBPRFX('40'), CMEBPFBA('10'), CMEBPFM3('02') AND | *MCM86178* |
| | * CMEBPXDE('01') USED IN CMEAFLG3 FOR MSG PREFIX INFO | *MCM86178* |
| 00000A | CMEPDFL3 DS X | THIRD PUT DETAIL FLAG |
| | * CMEBIFWD ('20'), CMEBFM3 ('08') AND CMEBXDES ('04') ADE | *JEB84057* |
| | * BITS ARE SET IN CMEPDFL4 WHEN PUT WITH MESSAGE SPECIFIED | *JEB84057* |
| 00000B | CMEPDFL4 DS X | FOURTH PUT DETAIL FLAG |
| 00000C | CMEPDSZ EQU *-*CME | SIZE BASIC PUT DETAIL CME |
| | ***** | *JEB84057* |
| | * AFTER THIS POINT ARE THE ADE'S AND RPE'S AS REQUIRED FOR | *JEB84057* |
| | * KEY AND MESSAGE PARAMETERS. | *JEB84057* |
| | ***** | *JEB84057* |
| | * | *JEB84057* |
| | * | *JEB84057* |
| | * GET DETAIL CME CLASS=4, FUNCTION=3 | *JEB84057* |
| | * | *JEB84057* |
| 00000C | ORG CMEFLAG1 | |

```

CMEGTFR #FLAG X'80'          GET DETAIL FIRST (VS NEXT)*LCB84082*
CMEGTMF #FLAG X'40'          GET DET MOD FIELDS(VS ALL)*LCB84082*
CMEGTRK #FLAG X'20'          GET DETAIL RET KEY (VS KEY*LCB84082*
CMEGTKY #FLAG X'10'          GET DETAIL KEY (VS RET KEY*LCB84082*
000008   CMEGTFL1 DS X        FIRST PAGE FLAG BYTE *JEB84057*
000009   CMEGTFL2 DS X        SECOND PAGE FLAG BYTE *JEB84057*
* CMEBIFWD ('20'), CMEBFM3 ('08') AND CMEBXDES ('04') ADE **LCB84082*
* BITS ARE SET IN CMEPDL3 WHEN PUT WITH KEY IS SPECIFIED **LCB84082*
00000A   CMEGTFL3 DS X        THIRD PAGE FLAG BYTE *JEB84057*
00000B   CMEGTFL4 DS X        FOURTH PAGE FLAG BYTE *JEB84057*
00000C   CMEGTSZ EQU *-*CME    SIZE OF GET DETAIL CME *JEB84057*
*****LCB84082*
* AFTER THIS POINT ARE THE ADE'S AND XDE'S AS REQUIRED FOR **LCB84082*
* A KEY PARAMETER (RETURN KEY, OR KEY) **LCB84082*
*****LCB84082*
*
*
* CLOSE FILE MAPS           CLASS=4, FUNCTION=4
*
00000C
      ORG  CMEFLAG1
      CMECLIN #FLAG X'80'      CLOSE INPUT
      CMECLOU #FLAG X'40'      CLOSE OUTPUT
      CMECLBO #FLAG X'10'      CLOSE BOTH
000008   CMECLFL1 DS X       FIRST CLOSE FILE MAP FLAG
000009   CMECLFL2 DS X       SECOND CLOSE FILE FLAG
00000A   CMECLFL3 DS X       THIRD CLOSE FILE FLAG
00000B   CMECLFL4 DS X       FOURTH CLOSE FILE FLAG
00000C   CMECLSZ EQU *-*CME  SIZE OF CLOSE FILE CME
*
*
* DC ACCEPT CME           CLASS=5, FUNCTION=1
*
00000C
      ORG  CMEFLAG1
      CMEAUTC #FLAG X'80'     ACCEPT TASK CODE
      CMEAUTI #FLAG X'40'     ACCEPT TASK ID
      CMEAULI #FLAG X'20'     ACCEPT LTERM ID
      CMEAUCPI #FLAG X'10'    ACCEPT PTERM ID
      CMEACSV #FLAG X'08'     ACCEPT SYSTEM VERSION
      CMEAUCUI #FLAG X'04'    ACCEPT USER ID
      CMEAUCSS #FLAG X'02'    ACCEPT SCREEN SIZE
      CMERUNP #FLAG X'01'     ACCEPT RUN PARAMETERS
000008   CMEAFCFL1 DS X      FIRST ACCEPT FLAG BYTE
000009   CMEAFCFL2 DS X      SECOND ACCEPT FLAG BYTE
      CMEAFCDD #FLAG X'80'    ACCEPT CURRENT DIALOG NAME
      CMEAFCHD #FLAG X'40'    ACCEPT NEXT HIGHER DIALOG NAME
      CMEAFCRC #FLAG X'20'    ACCEPT CURRENT RECORD NAME
      CMEAFCAR #FLAG X'10'    ACCEPT CURRENT AREA NAME
      CMEAFCES #FLAG X'08'    ACCEPT ERROR SET NAME
      CMEAFCER #FLAG X'04'    ACCEPT ERROR RECORD NAME
      CMEAFCEA #FLAG X'02'    ACCEPT ERROR AREA NAME
00000A   CMEAFCFL3 DS X      THIRD ACCEPT FLAG BYTE
      CMEAFCXD #FLAG X'40'    ACCEPT RUN PARMS ADE IS FOLLOWED BY
                                AN XDE STACK
00000B   CMEAFCFL4 DS X      FOURTH ACCEPT FLAG BYTE
00000C   CMEAFCSZ EQU *-*CME  SIZE OF BASIC DC ACCEPT CME
      CMEAFCXS DS 0F          XDE STACK(S) BEGIN HERE EXCEPT FOR
                                ACCEPT RUN PARMS WHICH HAS AN ADE
                                HERE
***
***
* PUT/GET/DELETE SCRATCH CME CLASS=5, FUNCTION=2
* PUT/GET/DELETE QUEUE CME   CLASS=5, FUNCTION=3
*
00000C
      ORG  CMEFLAG1
*
* THE FOLLOWING EQUATES ARE USED TO SPECIFY THE BIT SETTINGS FOR THE

```

* FIRST FIVE FLAGS AS OPPOSED TO #FLAG MACROS WHICH BIND THE
 * BIT SETTINGS TO A SPECIFIC FLAG.

*

| | | |
|--------|--|--|
| 00020 | CMESQFM1 EQU X'20' | ADE IS AN EMBEDDED FULLWORD |
| 00010 | CMESQFM2 EQU X'10' | ADE IS LITPOOL OFFSET & LENGTH |
| 00008 | CMESQFM3 EQU X'08' | ADE IS FORMAT 3 (4 HALFWORDS) |
| 00004 | CMESQXDE EQU X'04' | ADE IS FOLLOWED BY AN XDE STACK |
| 00002 | CMESQFIR EQU X'02' | XDE STACK MOVES DATA FROM THE IRA |
| 00001 | CMESQPTR EQU X'01' | XDE STACK SIMPLY POINTS TO A FIELD |
| | | ** IF SQXDE IS SET BUT NEITHER SQFIR NOR SQPTR IS SET, IT IS ASSUMED |
| | | ** THAT THE XDE STACK MOVES DATA TO THE IRA. |
| 000008 | CMESQFL1 DS X | SCRATCH/QUEUE FLAG ONE |
| 000009 | CMESQFL2 DS X | SCRATCH/QUEUE FLAG TWO |
| | CMESQTO #FLAG X'40' | ADE POINTS TO A "TO" FIELD FOR LEN |
| 00000A | CMESQFL3 DS X | SCRATCH/QUEUE FLAG THREE |
| 00000B | CMESQFL4 DS X | SCRATCH/QUEUE FLAG FOUR |
| 00000C | CMESQFL5 DS X | SCRATCH/QUEUE FLAG FIVE |
| | CMEMEC #FLAG X'80' | SCRATCH QUEUE OR PRINT USING |
| | "ALLOWING ERROR STATUS" CLAUSE | * |
| 00000D | CMESQFL6 DS X | SCRATCH/QUEUE FLAG SIX |
| | CMESQPT #FLAG X'01' | PUT SCRATCH OR QUEUE |
| | CMESQGT #FLAG X'02' | GET SCRATCH OR QUEUE |
| | CMESQDL #FLAG X'04' | DELETE SCRATCH OR QUEUE |
| | CMESQPR #FLAG X'08' | PRIOR RECORD SPECIFIED |
| | CMESQNX #FLAG X'10' | NEXT RECORD SPECIFIED |
| | CMESQRI #FLAG X'20' | CURRENT REC, REC ID OR SEQ # |
| | CMESQLA #FLAG X'40' | LAST RECORD SPECIFIED |
| | CMESQFI #FLAG X'80' | FIRST RECORD SPECIFIED |
| | CMESQAL #FLAG X'F8' | ALL SPECIFIED ON DEL SCRATCH/QUEUE |
| | CMESQRE #FLAG X'F9' | REPLACE SPECIFIED ON PUT SCRATCH |
| 00000E | CMESQSQ1 DS X | SCRATCH/QUEUE PARAMETER FLAG ONE |
| | CMESQRA #FLAG X'1F' | RETURN TO ADS FOR ALL CONDITIONS |
| | CMESQRT #FLAG X'20' | RETENTION SPECIFIED ON PUT QUEUE |
| | CMESQWA #FLAG X'40' | WAIT SPECIFIED ON GET QUEUE |
| | CMESQDG #FLAG X'80' | NOLOCK SPECIFIED ON GET QUEUE |
| 00000F | CMESQSQ2 DS X | SCRATCH/QUEUE PARAMETER FLAG TWO |
| 00010 | CMESQSZ EQU *-CME | SIZE OF BASIC SCRATCH/QUEUE CME |
| 000010 | CMESQADE DS 0F | START OF ARGUMENT DESCRIPTION |
| | * | |
| | * | |
| | * | |
| | * WRITE PRINTER | CLASS=5, FUNCTION=4 |
| | * | |
| 000010 | ORG CMEFLAG1 | |
| | * | |
| | * | |
| | * | |
| | * THE FOLLOWING EQUATES ARE USED TO SPECIFY THE BIT SETTINGS FOR THE | |
| | * FIRST FIVE FLAGS AS OPPOSED TO #FLAG MACROS WHICH BIND THE | |
| | * BIT SETTINGS TO A SPECIFIC FLAG. | |
| | * | |
| 00020 | CMEWPFM1 EQU X'20' | ADE IS AN EMBEDDED FULLWORD |
| 00010 | CMEWPFM2 EQU X'10' | ADE IS LITPOOL OFFSET & LENGTH |
| 00008 | CMEWPFM3 EQU X'08' | ADE IS FORMAT 3 (4 HALFWORDS) |
| 00004 | CMEWPXDE EQU X'04' | ADE IS FOLLOWED BY AN XDE STACK |
| 00001 | CMEWPPTR EQU X'01' | XDE STACK SIMPLY POINTS TO A FIELD |
| | ** IF WPXDE IS SET BUT WPPTR IS NOT SET, IT IS ASSUMED | |
| | ** THAT THE XDE STACK MOVES DATA TO THE IRA. | |
| 000008 | CMEWPFL1 DS X | WRITE PRINTER FLAG ONE |
| | CMEWPTO #FLAG X'40' | ADE POINTS TO A "TO" FIELD FOR LEN |
| 000009 | CMEWPFL2 DS X | WRITE PRINTER FLAG TWO |
| 00000A | CMEWPFL3 DS X | WRITE PRINTER FLAG THREE |
| 00000B | CMEWPFL4 DS X | WRITE PRINTER FLAG FOUR |
| 00000C | CMEWPFL5 DS X | WRITE PRINTER FLAG FIVE |
| | CMEWPKP #FLAG X'04' | KEEP SPECIFIED *JEB84257* |
| | CMEWPHD #FLAG X'02' | HOLD SPECIFIED *JEB84257* |
| | CMEWPAL #FLAG X'01' | ALL SPECIFIED *JEB84257* |
| 00000D | CMEWPFL6 DS X | WRITE PRINTER FLAG SIX (X'80' BIT) |

```

*
*
CMEWPNP #FLAG X'01' ALSO USED. SAME AS CMEMEC FLAG
CMEWPNM #FLAG X'02' ON SCRATCH/QUEUE CME)
CMEWPER #FLAG X'04' NEWPAGE OR ERASE
CMEWPCD #FLAG X'08' NATIVE MODE DATA STREAM
CMEWPBS #FLAG X'10' ENDRPT (END OF REPORT)
CMEWPSC #FLAG X'20' USE DEFAULT CLASS OR DEST.
CMEWPUD #FLAG X'40' USE DEFAULT BUFSIZE (ALWAYS SET)
CMEWPBT #FLAG X'80' PRINT SCREEN CONTENTS
00000E     CMEWPWP1 DS   X USE DESTINATION, NOT CLASS
            CMEWPRA #FLAG X'3F' DC BATCH (NEVER SET)
            CMEWPCS #FLAG X'40' WRITE PRINTER PARAMETER FLAG ONE
00000F     00010  CMEWPWP2 DS   X RETURN TO ADS FOR ALL CONDITIONS
            CMEWPSZ EQU  *--CME COPIES WAS SPECIFIED
            CMEWPADE DS   OF WRITE PRINTER PARAMETER FLAG TWO
000010
            *
            *
            *
            * INIT RECORDS CME      CLASS=5,  FUNCTION=5
            *
000010     ORG   CMEFLAG1
            CMEINTA #FLAG X'80' INIT ALL RECORDS
000008     CMEIFLG1 DS   X FIRST INIT FLAG BYTE
000009     CMEIFLG2 DS   X SECOND INIT FLAG BYTE
00000A     CMEIFLG3 DS   X THIRD INIT FLAG BYTE
00000B     CMEIFLG4 DS   X FOURTH INIT FLAG BYTE
00000C     CMEINTSZ EQU  *--CME SIZE OF BASIC INIT REC CME
00000C     CMEIRCCT DS   H NUMBER OF RECORDS TO INITIALIZE
00000E     CMEIROFF DS   OH START OF VRE INDICES (ONE HALFWORD
            *
            *
            *
            * KEEP LONGTERM CME    CLASS=5,  FUNCTION=6
            *
00000E     ORG   CMEFLAG1
            CMEKNTF #FLAG X'80' NOTIFY CURRENT
            CMEKLOC #FLAG X'40' LOCK CURRENT (SHARE OR EXCLUSIVE)
            CMEKUPG #FLAG X'20' UPGRADE
            CMEKTST #FLAG X'10' TEST
            CMEKRLS #FLAG X'08' RELEASE
            CMEKXCL #FLAG X'04' ON=EXCLUSIVE, OFF=SHARE
            CMEKNWT #FLAG X'02' NO WAIT
            CMEKNDL #FLAG X'01' NO DEADLOCK
000008     CMEKFLG1 DS   X FIRST KEEP LONGTERM FLAG BYTE
            CMEKAD1 #FLAG X'80' FIRST ADE IS USED
            CMEKIDX #FLAG X'40' FIRST ADE IS AN RDE INDEX
            CMEKSSA #FLAG X'20' FIRST ADE IS A SSAN TBL OFFSET
000009     CMEKFLG2 DS   X SECOND KEEP LONGTERM FLAG BYTE
            CMEKXDE #FLAG X'80' ADE FOR LONGTERM ID HAS AN XDE STACK
            CMEKRTL #FLAG X'08' ADE FOR RETURN-LOCATION IS USED
00000A     CMEKFLG3 DS   X THIRD KEEP LONGTERM FLAG BYTE
            ** CMEBEC FLAG IS SET IN CMEKFLG4 FOR ALLOWING CLAUSE EXISTENCE
00000B     CMEKFLG4 DS   X FOURTH KEEP LONGTERM FLAG BYTE
00000C     CMEKADES DS   F FMT 1 OR 2 ADE FOR REC/SET/AREA DESC
00010      00010  CMEKLKSZ EQU  *--CME SIZE OF BASIC KEEP LONGTERM CME
            CMEKLTID DS   OH START OF ADE FOR KEEP LONGTERM ID
            *
            *
            * SNAP DUMP CME    CLASS=5,  FUNCTION=7
            *
000010     ORG   CMEFLAG1
            CMENOCB #FLAG X'80' ON IF SNAP OCB SPECIFIED
            CMENOWA #FLAG X'40' ON IF SNAP OWA SPECIFIED
            CMENOTB #FLAG X'20' ON IF SNAP OTB SPECIFIED
            CMENFDB #FLAG X'10' ON IF SNAP FDB SPECIFIED

```

| | | |
|--------|---|--|
| | CMENVDB #FLAG X'08' | ON IF SNAP VDB CHAIN SPECIFIED |
| | CMENRBB #FLAG X'04' | ON IF SNAP RBB CHAIN SPECIFIED |
| | CMENRGS #FLAG X'02' | ON IF SNAP REGISTERS SPECIFIED |
| 000008 | CMENFLG1 DS X | FIRST SNAP FLAG BYTE |
| | CMENADB #FLAG X'80' | ON IF SNAP ADB SPECIFIED *JEB84115* |
| | CMENTAT #FLAG X'40' | ON IF SNAP TAT SPECIFIED *JEB84115* |
| | CMENOTX #FLAG X'20' | ON IF SNAP OTB EXTENSION *JEB84115* |
| | CMENREX #FLAG X'10' | ON IF SNAP REX SPECIFIED *JMA86230* |
| 000009 | CMENFLG2 DS X | SECOND SNAP FLAG BYTE |
| 00000A | CMENFLG3 DS X | THIRD SNAP FLAG BYTE |
| 00000B | CMENFLG4 DS X | FOURTH SNAP FLAG BYTE |
| 00000C | CMENTTLN DS H | LENGTH OF TITLE LITERAL (ZERO IF NO TITLE LITERAL) |
| | * | |
| 00000E | CMENTTDF DS H | OFFSET INTO LITERAL POOL FOR TITLE |
| 000010 | CMENRCCT DS H | NUMBER OF RECORDS TO BE SNAPPED |
| 00012 | CMENSNSZ EQU *-CME | SIZE OF BASIC SNAP CME |
| 000012 | CMENROFF DS OH | START OF VRE INDICES (ONE HALFWORD FOR EACH RECORD TO BE SNAPPED) |
| | * | |
| | * | |
| | * | |
| | * COMMIT TASK CME CLASS=5, FUNCTION=8 | |
| | * ROLLBACK TASK CME CLASS=5, FUNCTION=9 | |
| | * | |
| 000012 | ORG CMEFLAG1 | |
| 000008 | CMECTAL #FLAG X'80' | ON IF COMMIT TASK ALL SPECIFIED |
| 000009 | CMETSKF1 DS X | 1ST COMMIT/ROLLBACK TASK FLAG BYTE |
| 00000A | CMETSKF2 DS X | SECOND COMMIT/RBCK TASK FLAG BYTE |
| 00000B | CMETSKF3 DS X | THIRD COMMIT/RBCK TASK FLAG BYTE |
| 00000C | CMETSKF4 DS X | FOURTH COMMIT/RBCK TASK FLAG BYTE |
| 00000C | CMETSKSZ EQU *-CME | SIZE OF BASIC COMMIT/RBCK TASK CME |
| | * | |
| | * | |
| | * | |
| | * EXECUTE NEXT FUNCTION CME CLASS=5, FUNCTION=10 | |
| | * | |
| 00000C | ORG CMEFLAG1 | *JMA85219* |
| | CMEENN #FLAG X'80' | NOSAVE INDICATOR **JMA85219* |
| | CMEENN #FLAG X'40' | NOFINISH INDICATOR **JMA85219* |
| 000008 | CMEENFL1 DS X | EXEC NEXT FUNC FLAG 1 **JMA85219* |
| 000009 | CMEENFL2 DS X | EXEC NEXT FUNC FLAG 2 **JMA85219* |
| 00000A | CMEENFL3 DS X | EXEC NEXT FUNC FLAG 3 **JMA85219* |
| 00000B | CMEENFL4 DS X | EXEC NEXT FUNC FLAG 4 **JMA85219* |
| 00000C | CMENFLEN EQU *-CME | LENGTH OF EXECUTE NEXT FUNC CME |
| | * | *JMA85058* |
| | * | *JMA85058* |
| | * WRITE TO LOG/OPERATOR CME CLASS=5, FUNCTION=11 | *DKJ85135* |
| | * | *JMA85058* |
| 00000C | ORG CMEFLAG1 | *JMA85058* |
| | CMEWLOG #FLAG X'80' | WRITE TO LOG FLAG *DKJ85135* |
| | CMEWOPR #FLAG X'40' | WRITE TO OPER FLAG *DKJ85135* |
| | CMEWLM #FLAG X'20' | MESSAGE FLAG *DKJ85135* |
| 000008 | CMEWLFL1 DS X | FIRST WRITE LOG FLAG *JMA85058* |
| | * CMEDISP ('80') AND CMEDISL ('20') FROM DISPLAY ARE USED FOR FLAGS | |
| | * ALSO CMEDSMS ('40') | |
| 000009 | CMEWLFL2 DS X | SECOND WRITE LOG FLAG *JMA85058* |
| | * CMEBPRFX('40'), CMEBPFBA('10'), CMEBPFM3('02') AND | *MCM86178* |
| | * CMEBPXDE('01') USED IN CMEAFLG3 FOR MSG PREFIX INFO | *MCM86178* |
| 00000A | CMEWLFL3 DS X | THIRD WRITE LOG FLAG *JMA85058* |
| 00000B | CMEWLFL4 DS X | FOURTH WRITE LOG FLAG *JMA85058* |
| 00000C | CMEWLSZ EQU *-CME | SIZE OF BASIC WRTLOG CME *JMA85058* |
| 00000C | CMEWLLSZ DS H | LIT SIZE FOR MESSAGE *JMA85058* |
| 00000E | CMEWLLOF DS H | LIT OFFSET IN LIT POOL *JMA85058* |
| 000010 | ORG CMEWLLSZ | *JMA85058* |
| 00000C | CMEWLLKY DS F | PRESTORED MESSAGE CODE *JMA85058* |
| 000004 | CMEWLLMSZ EQU *-CMEWLLSZ | BASIC MSG DESCRIPT SIZE *JMA85058* |
| 000010 | CMEWLLIT DS OH | START OF INLINE LITERAL *JMA85058* |

```

000010      CMEWLMPN DS   F          NUM OF MSG REPLCMENT PARM*JMA85058*
000014      CMEWLRPL DS  OF         START OF REP PARM LIST *JMA85058*
*****JMA85058*
* IF REPLACEMENT PARMS ARE SPECIFIED, THERE WILL BE ONE REPLACEMENT *
* PARAMETER ELEMENT (RPE) HERE FOR EACH PARM (SEE RPE DSECT)      *
*
*
*
*      TRACE  FUNCTION CME          CLASS=5,    FUNCTION=12
*
000014      ORG    CMEFLAG1          *JMA85219*
      CMETOFF #FLAG X'80'        TRACE OFF INDICATOR  **JMA91057*
      CMETALL #FLAG X'40'        TRACE ALL INDICATOR  **JMA91057*
      CMETCTL #FLAG X'20'        TRACE CTL INDICATOR  **JMA91057*
000008      CMETFLG1 DS   X        TRACE FUNCTION FLAG 1 **JMA91057*
000009      CMETFLG2 DS   X        TRACE FUNCTION FLAG 2 **JMA91057*
00000A      CMETFLG3 DS   X        TRACE FUNCTION FLAG 3 **JMA91057*
00000B      CMETFLG4 DS   X        TRACE FUNCTION FLAG 4 **JMA91057*
00000C      CMETLEN EQU  *-*CME      LENGTH OF TRACE FUNCTION CME
*****
*
*      CHECKPOINT FUNCTION CME      CLASS=6,    FUNCTION=1      *JMA85058**
*
00000C      ORG    CMEBODY          *LCB85039*
00000C      CMECKLEN EQU  *-*CME      LENGTH OF CHECKPOINT CME *LCB85039*
*
*****
*
*      SEND/RECEIVE CME'S          CLASS=7
*
*      ALLOCATE CME              CLASS=7    FUNCTION=1
*
00000C      ORG    CMEFLAG1          SECURITY = SAME
      CMEALSS #FLAG X'80'        SECURITY = PGM
      CMEALSP #FLAG X'40'        SYNC-LEVEL = CONFIRM
      CMEALSC #FLAG X'20'        NOFORMAT
      CMEALNF #FLAG X'10'        LU-NAME IS LIT (FORMAT 2 ADE)
      CMEALLS #FLAG X'08'        MODE-NAME IS LIT (FORMAT 2 ADE)
      CMEALMS #FLAG X'04'        TPN IS LIT (FORMAT 2 ADE)
      CMEALTS #FLAG X'02'        USER-ID IS LIT (FORMAT 2 ADE)
      CMEALUS #FLAG X'01'        FLAG 1
000008      CMEAFL1 DS   X        LU-NAME FMT 3 ADE W/ XDE
      CMEAFLX #FLAG X'80'        MODE-NAME FMT 3 ADE W/ XDE
      CMEAFLM #FLAG X'40'        TPN FMT 3 ADE W/ XDE
      CMEAFLT #FLAG X'20'        USER-ID FMT 3 ADE W/XDE
      CMEAFLU #FLAG X'10'        PASSWORD IS LIT (FORMAT 2 ADE)
      CMEAFLP #FLAG X'08'        PASSWORD FMT 3 ADE W/ XDE
      CMEAFLV #FLAG X'04'        LOCAL VER LIT (FORMAT 2 ADE)
      CMEAFLV #FLAG X'02'        LOCAL VER PRESENT
      CMEAFLV #FLAG X'01'        FLAG 2
000009      CMEAFL2 DS   X        LOCAL VER FMT 3 ADE W/ XDE
      CMEAFLV #FLAG X'80'        REMOTE VER FMT 3 ADE W/ XDE
      CMEAFLV #FLAG X'40'        REMOTE VER PRESENT
      CMEAFLV #FLAG X'20'        MODE-NAME PRESENT
      CMEAFLV #FLAG X'10'        OPEN NO FORMAT      **GWG**
      CMEAFLV #FLAG X'08'        NO LU-NAME PRESENT  **GWG**
      CMEAFLV #FLAG X'04'        NO DESTINATION PRESENT **GWG**
      CMEAFLV #FLAG X'02'        FLAG 3
00000A      CMEAFL3 DS   X        CME ADSO BUILT
      CMEAFLB #FLAG X'80'        FLAG 4
00000B      CMEAFL4 DS   X        LENGTH OF ALLOCATE CME
00000C      CMEALSZ EQU  *-*CME      OFFSET TO FD/FDE'S
00000C      CMESROF DS   F        LENGTH OF SEND/RECEIVE CME
00010       CMESRSZ EQU  *-*CME

```

PRINT GEN

1.37 #CMEEQU

```

#CMEEQU

*
*      EQUATES FOR ADSO COMMAND CLASSES AND FUNCTIONS
*
*
*      DATABASE COMMAND EQUATES
*
Offset  Value
00001  DBCLASS   EQU   1          DATABASE CLASS
*
00063  LRVFUNC   EQU   99         LOGICAL RECORD VERB FUNCTION
*
*      CONTROL COMMAND EQUATES
*
00002  CONCLASS  EQU   2          CONTROL CLASS
*
00000  ABRTFUNC  EQU   0          ABORT FUNCTION
00001  INVKFUNC  EQU   1          INVOKE FUNCTION
00002  TRANFUNC  EQU   2          TRANSFER FUNCTION
00003  RETFUNC   EQU   3          RETURN FUNCTION
00004  DISPFUNC  EQU   4          DISPLAY FUNCTION
00006  LEAVFUNC  EQU   6          LEAVE FUNCTION
00007  LINKFUNC  EQU   7          LINK FUNCTION
00008  WRTFUNC   EQU   8          WRITE TRANSACTION FUNC *JMA85058*
00009  READFUNC  EQU   9          READ TRANSACTION FUNC *JMA85058*
0000A  CONTFUNC  EQU   10         CONTINUE FUNCTION *JMA85058*
0000B  SQLFUNC   EQU   11         @SQLCALL FUNCTION      **SQL**
*
*      PROCEDURE COMMAND EQUATES
*
00003  PROCLASS  EQU   3          PROCEDURE CLASS
*
00001  ASGNFUNC  EQU   1          ASSIGNMENT FUNCTION
00002  CONDFUNC  EQU   2          CONDITIONAL FUNCTION
00003  REPWFUNC  EQU   3          REPEAT/WHILE FUNCTION
00004  BRCHFUNC  EQU   4          BRANCH FUNCTION
00005  SUBCFUNC  EQU   5          SUBROUTINE CALL FUNCTION
00006  SUBEFUNC  EQU   6          SUBROUTINE EXIT FUNCTION
00007  ONFUNC    EQU   7          ON PATH STATUS FUNCTION
00015  ADDFUNC   EQU   21         ASSIGNMENT/ADD FUNCTION
00016  SUBFUNC   EQU   22         ASSIGNMENT/SUBTRACT FUNCTION
00017  MULTFUNC  EQU   23         ASSIGNMENT/MULTIPLY FUNCTION
00018  DIVFUNC   EQU   24         ASSIGNMENT/DIVIDE FUNCTION
00019  DIVRFUNC  EQU   25         ASSIGNMENT/DIVIDE WITH REMAINDER
00028  MOVEFUNC  EQU   40         MOVE FUNCTION
00029  COMPFUNC  EQU   41         COMPUTE FUNCTION
*
*      MAP COMMAND EQUATES
*
00004  MAPCLASS  EQU   4          MAP CLASS
*
00001  MMAPFUNC  EQU   1          MODIFY MAP FUNCTION
00002  PUTDFUNC  EQU   2          PUT PAGEABLE MAP DETAIL *JEB84058*
00003  GETDFUNC  EQU   3          GET PAGEABLE MAP DETAIL *JEB84058*
00004  CMAPFUNC  EQU   4          CLOSE FILE MAP FUNCTION
*
*      MISCELLANEOUS COMMAND EQUATES
*
00005  MISCLASS  EQU   5          MISCELLANEOUS CLASS
*

```

| | | | | |
|--------------------------------|-----------|-----|----|-------------------------------------|
| 00001 | ACPTFUNC | EQU | 1 | ACCEPT FUNCTION |
| 00002 | SCRFUNC | EQU | 2 | SCRATCH FUNCTIONS |
| 00003 | QUEFUNC | EQU | 3 | QUEUE FUNCTIONS |
| 00004 | WPRTFUNC | EQU | 4 | WRITE PRINTER FUNCTION |
| 00005 | INITFUNC | EQU | 5 | INITIALIZE RECORDS FUNCTION |
| 00006 | KEPLFUNC | EQU | 6 | KEEP LONGTERM FUNCTION |
| 00007 | SNAPFUNC | EQU | 7 | SNAP FUNCTION |
| 00008 | CMTTFUNC | EQU | 8 | COMMIT TASK FUNCTION |
| 00009 | RBKTFUNC | EQU | 9 | ROLLBACK TASK FUNCTION |
| 0000A | EXENFUNC | EQU | 10 | EXECUTE NEXT FUNCTION FUNCTION |
| 0000B | WRTLFUNC | EQU | 11 | WRITE LOG FUNCTION *JMA85058* |
| 0000C | TRACFUNC | EQU | 12 | ONLINE TRACE FUNCTION *JMA91057* |
| * | | | | |
| * CICS COMMANDS EQUATES | | | | |
| * | | | | |
| 00006 | VSMCLASS | EQU | 6 | CICS CLASS |
| * | | | | |
| 00001 | READFNCT | EQU | 1 | READ FUNCTION |
| 00002 | WRITFUNC | EQU | 2 | WRITE FUNCTION |
| 00003 | REWRFUNC | EQU | 3 | REWRITE FUNCTION |
| 00004 | DELT FUNC | EQU | 4 | DELETE FUNCTION |
| 00005 | UNLK FUNC | EQU | 5 | UNLOCK FUNCTION |
| 00006 | STRBFUNC | EQU | 6 | START BROWSE FUNCTION |
| 00007 | RDNXFUNC | EQU | 7 | READ NEXT FUNCTION |
| 00008 | RDPRFUNC | EQU | 8 | READ PREVIOUS FUNCTION |
| 00009 | RSTBFUNC | EQU | 9 | RESET BROWSE FUNCTION |
| 0000A | ENDBFUNC | EQU | 10 | END BROWSE FUNCTION |
| 0000B | WRTDFUNC | EQU | 11 | WRITE TRANSIENT DATA FUNCTION |
| 0000C | RDTDFUNC | EQU | 12 | READ TRANSIENT DATA FUNCTION |
| 0000D | DLTDFUNC | EQU | 13 | DELETE TRANSIENT DATA FUNCTION |
| 0000E | WRTSFUNC | EQU | 14 | WRITE TEMP STORAGE FUNCTION |
| 0000F | RDTSFUNC | EQU | 15 | READ TEMP STORAGE FUNCTION |
| 00010 | DLTSFUNC | EQU | 16 | DELETE TEMP STORAGE FUNCTION |
| 00011 | SNCPFUNC | EQU | 17 | SYNCPOINT FUNCTION |
| * | | | | *JMA85058* |
| * CHECKPOINT COMMAND EQUATES | | | | *JMA85058* |
| * | | | | *JMA85058* |
| 00006 | CHKCLASS | EQU | 6 | DO WE USE IT?? CHECKPOINT CLASS |
| * | | | | *JMA85058* |
| 00001 | CHKPFUNC | EQU | 1 | DO WE NEED IT?? CHECKPOINT FUNCTION |
| * | | | | *JMA85058** |
| * SEND/RECEIVE COMMAND EQUATES | | | | *JMA85058* |
| * | | | | |
| 00007 | SRCLASS | EQU | 7 | SEND/RECEIVE CLASS |
| * | | | | |
| 00001 | ALLOFUNC | EQU | 1 | ALLOCATE FUNCTION |
| 00002 | CTSSFUNC | EQU | 2 | CONTROL SESSION FUNCTION |
| 00003 | SDDAFUNC | EQU | 3 | SEND-DATA FUNCTION |
| 00004 | CNFMFUNC | EQU | 4 | CONFIRM FUNCTION |
| 00005 | CNFDFUNC | EQU | 5 | CONFIRMED FUNCTION |
| 00006 | RQSDFUNC | EQU | 6 | REQUEST-TO-SEND FUNCTION |
| 00007 | SDERFUNC | EQU | 7 | SEND-ERROR FUNCTION |
| 00008 | RCWTFUNC | EQU | 8 | RECEIVE-AND-WAIT FUNCTION |
| 00009 | PRRCFUNC | EQU | 9 | PREPARE-TO-RECEIVE FUNCTION |
| 0000A | DEALFUNC | EQU | 10 | DEALLOCATE FUNCTION |
| * | | | | *JMA85058* |
| * OSCAR COMMAND EQUATES | | | | |
| * | | | | |
| 00008 | OSCLASS | EQU | 8 | SEND/RECEIVE CLASS |
| * | | | | |
| 00001 | OPENFUNC | EQU | 1 | OPEN FUNCTION |
| 00002 | CLOSFUNC | EQU | 2 | CLOSE FUNCTION |
| 00003 | SENDFUNC | EQU | 3 | SEND FUNCTION |
| 00004 | RECVFUNC | EQU | 4 | RECEIVE FUNCTION |

1.38 #DGBDS

```

COPY #DGBDS
*****
***          DGB : ADSO DIALOG GENERATION BLOCK
***          DGB IS A DSECT THAT DEFINES THE DIALOG GENERATION BLOCK
***          USED BY THE ADS ONLINE GENERATOR. IT IS A FIXED LENGTH
***          PIECE OF KEPT STORAGE THAT CONTAINS THE BASIC DIALOG DATA
***          NEEDED TO REESTABLISH THE GENERATION OF A DIALOG AFTER
***          ANY BREAK IN PROCESSING (THAT IS, AFTER A CONVERSE, SYSTEM
***          CRASH, OR USER SUSPEND).
***          THE DGB IS ALSO USED BY THE ADS/ONLINE APPLICATION
***          GENERATOR BECAUSE OF THE STORAGE MANAGEMENT INFORMATION
***          WHICH IS MAINTAINED IN IT BY ADSOGNRC.
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--------------|
| 000000 | DGB | DSECT |
| 000000 | DGBID | DS CL4 |
| 000004 | DGBDNAM | DS CL8 |
| 00000C | DGBDVER | DS H |
| 00000E | | ORG DGBDNAM |
| 000004 | DGBANAM | DS CL8 |
| 00000C | DGBAVER | DS H |
| 00000E | DGBFSAFE | DS H |
| 000010 | DGBCURMD | DS H |
| 000012 | DGBCURFC | DS H |
| 000014 | DGBFGWAA | DS A |
| 000018 | DGBLGWAA | DS A |
| 00001C | DGBFPWAA | DS A |
| 000020 | DGBLPWAA | DS A |
| 000024 | DGBSRTA | DS A |
| 000028 | | ORG DGBSRTA |
| 000024 | DGBFDWAA | DS A |
| 000028 | DGBIRECA | DS A |
| 00002C | DGBTHRSH | DS F |
| 000030 | DGBQSIZE | DS F |
| 000034 | | ORG DGBQSIZE |
| 000030 | DGBLDWAA | DS A |
| 000034 | DGBPWKSZ | DS F |
| 000038 | DGBSWKSZ | DS F |
| 00003C | DGBIRECL | DS F |
| 000040 | DGBNSGWA | DS H |
| 000042 | DGBNSPWA | DS H |
| 00044 | DGBALEN | EQU *-DGB |
| 000044 | DGBLSTMD | DS H |
| 000046 | DGBLSTFC | DS H |
| | DGBSCR | #FLAG X'80' |
| 000048 | DGBSCRI | DS 0XL1 |
| 00080 | DGBSCRM | EQU X'80' |
| 000048 | DGBFLAG1 | DS X |
| | DGBAST | #FLAG X'80' |
| 000049 | DGBASTI | DS 0XL1 |
| 00080 | DGBASTM | EQU X'80' |
| | DGBASTM | #FLAG X'40' |
| 000049 | DGBASTMI | DS 0XL1 |
| 00040 | DGBASTMM | EQU X'40' |
| | DGBSDRM | #FLAG X'20' |
| 000049 | DGBSDRMI | DS 0XL1 |
| 00020 | DGBSDRMM | EQU X'20' |

| | | | |
|--------|----------|--------------------|--------------------------------------|
| 000049 | DGBACT | #FLAG X'10' | ON IF ACTIVITY LOGGING ENABLED |
| | DGBACTI | DS 0XL1 | |
| 00010 | DGBACTM | EQU X'10' | |
| | DGBCOBM | #FLAG X'08' | ON IF COBOL MOVES ENABLED *LCB84100* |
| 000049 | DGBCOBMI | DS 0XL1 | |
| | 00008 | DGBCOBMM EQU X'08' | |
| 000049 | DGBFLAG2 | DS X | |
| 00004A | DGBSDRV | DS H | AUTOSTATUS RECORD VERSION |
| 00004C | DGBSDRNM | DS CL32 | AUTOSTATUS RECORD NAME |
| 0006C | DGBLEN | EQU *-DGB | LENGTH OF DGB |

1.39 #DIBDS

```

COPY #DIBDS
*****
***          DIB: DIALOG INFORMATION BLOCK (ADS RUNTIME) ***
***          DIB IS A DSECT THAT DESCRIBES THE CONTROL BLOCK CONTAINING ***
***          THE DIALOG INFORMATION TABLE ENTRIES (DIT); THIS SPACE      ***
***          MANAGEMENT CONTROL BLOCK IS USED TO PREVENT OVERLY FRAGMENTED ***
***          STORAGE.                                              ***
***          RESERVED
***          LENGTH OF DIB HEADER INFORMATION
***          NEXT EQUATE USED FOR ALLOCATION PURPOSES; IT IS AN ARBITRARY VALUE
***          # OF LONG ENTRIES IN CNTROL BLOCK
***          DIT: DIALOG INFORMATION TABLE (ADS RUNTIME)
***          DIT IS A DSECT THAT DESCRIBES TABLE ENTRIES WITH DIALOG NAMES ***
***          AND THEIR ASSOCIATED MRB/FILE OVERRIDE/STATISTICS/ERROR COUNT ***
***          INFORMATION.
***          THIS SHORT PORTION OF THE DSECT IS USED WHEN THE DIALOG
***          INFORMATION TABLE ENTRIES ARE CREATED AT RUNTIME. NO
***          OVERRIDES CAN BE ENTERED AT THIS POINT, SO ONLY THE BASIC
***          DIALOG INFORMATION IS NECESSARY.
***          DIALOG NAME
***          DIALOG'S INPUT MRB ADDRESS
***          DIALOG'S OUTPUT MRB ADDRESS
***          DIALOG STATISTICS BEING COLLECTED
***          EXTENDED (LONG) FORM OF DSECT USED
***          DIALOG HAS BEEN USED
***          EOF HIT ON INPUT      *JEB86051*
***          FLAG BYTE
***          DIALOG TRACE ON      *DKJ86008*
***          DITTONI DS 0XL1
***          DITTOMN EQU X'80'

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------------|------------------------------------|
| 000000 | DIB DSECT | 10:36:24 02/20/86 |
| 000000 | DIBID DS CL4 | DIB* IDENTIFICATION LABEL |
| 000004 | DIBNEXTA DS A | ADDR OF NEXT DIB CONTROL BLOCK |
| 000008 | DIBFRSPC DS H | FREE SPACE IN THIS CONTROL BLOCK |
| 00000A | DS H | RESERVED |
| 00000C | DIBNENT DS H | # OF USED ENTRIES IN THIS BLOCK |
| 00000E | DS H | RESERVED |
| 00010 | DIBLEN EQU ((*-DIB+3)/4)*4 | LENGTH OF DIB HEADER INFORMATION |
| 0000F | DIBTENT EQU 15 | # OF LONG ENTRIES IN CNTROL BLOCK |
| 000000 | DIT DSECT | |
| 000000 | DITDLGNM DS CL8 | DIALOG NAME |
| 000008 | DITIMRBA DS A | DIALOG'S INPUT MRB ADDRESS |
| 00000C | DITOMRBA DS A | DIALOG'S OUTPUT MRB ADDRESS |
| 000010 | DITSTAT DS 0XL1 | DIALOG STATISTICS BEING COLLECTED |
| 000080 | DITSTATM EQU X'80' | |
| 000010 | DITLONG DS 0XL1 | EXTENDED (LONG) FORM OF DSECT USED |
| 000040 | DITLONGGM EQU X'40' | |
| 000010 | DITREF DS 0XL1 | DIALOG HAS BEEN USED |
| 000020 | DITREFM EQU X'20' | |
| 000010 | DITEOF DS 0XL1 | EOF HIT ON INPUT *JEB86051* |
| 000010 | DITEOFI DS 0XL1 | |
| 000010 | DITEOFM EQU X'10' | |
| 000010 | DITFLAG1 DS X | FLAG BYTE |
| 000011 | DITTON DS 0XL1 | DIALOG TRACE ON *DKJ86008* |
| 000080 | DITTOMN EQU X'80' | |

| | | | |
|---|-----------------------------|-----------------------------------|------------|
| | DITTSUB #FLAG X'40' | DIALOG TRACE SUBROUTINES | *DKJ86008* |
| 000011 | DITTSUBI DS OXL1 | | |
| 00040 | DITTSUBM EQU X'40' | | |
| | DITTCMD #FLAG X'20' | DIALOG TRACE COMMANDS | *DKJ86008* |
| 000011 | DITTCMDI DS OXL1 | | |
| 00020 | DITTCMDM EQU X'20' | | |
| | DITCTD #FLAG X'08' | DIALOG CURRENT TRACE ON | *DKJ86008* |
| 000011 | DITCTDI DS OXL1 | | |
| 00008 | DITCTDM EQU X'08' | | |
| | DITCTS #FLAG X'04' | DIALOG CURRENT TRACE SUBS | *DKJ86008* |
| 000011 | DITCTS1 DS OXL1 | | |
| 00004 | DITCTS1M EQU X'04' | | |
| | DITCTC #FLAG X'02' | DIALOG CURRENT TRACE CMDS | *DKJ86008* |
| 000011 | DITCTCI DS OXL1 | | |
| 00002 | DITCTCM EQU X'02' | | |
| 000011 | DITFLAG2 DS X | FLAG BYTE | *DKJ86008* |
| 000012 | DITTBDF DS H | TRACE DLG BEFORE COUNT | *DKJ86008* |
| 000014 | DITTAFT DS H | TRACE DLG AFTER COUNT | *DKJ86008* |
| 000016 | DITTEVE DS H | TRACE DLG EVERY COUNT | *DKJ86008* |
| 000018 | DITEXEC DS F | EXECUTION COUNT | *DKJ86008* |
| 00001C | DS F | RESERVED | *DKJ86008* |
| 00020 | DITSLEN EQU ((*-DIT+3)/4)*4 | LENGTH OF SHORT DIT ENTRY | |
| ***** | | | |
| ***** THIS EXTENDED PORTION OF THE DSECT IS ONLY USED WHEN THE | | | |
| ***** DIALOG ENTRY IS BUILT DURING SYSIN DATASET PROCESSING. | | | |
| ***** THAT WOULD BE THE ONLY TIME THAT THE FOLLOWING PARAMETERS | | | |
| ***** COULD POSSIBLY BE DETERMINED. | | | |
| ***** | | | |
| 000020 | DITFILE1 DS OCL10 | | |
| 000020 | DITFILE DS CL8 | OS INPUT FILE NAME OVERRIDE | |
| 000028 | DS CL2 | FILLER | |
| 00002A | ORG DITFILE1 | | |
| 000020 | DITFILED DS CL7 | DOS INPUT FILE NAME OVERRIDE | |
| 000027 | DITINUMD DS CL3 | DOS INPUT FILE SYSNUM OVERRIDE | |
| 00002A | ORG | | |
| 00002A | DITIDTYP DS CL4 | DOS INPUT FILE DEVICE TYPE | |
| 00002E | DITFILE2 DS OCL10 | | |
| 00002E | DITOFILE DS CL8 | OS OUTPUT FILE NAME OVERRIDE | |
| 000036 | DS CL2 | FILLER | |
| 000038 | ORG DITFILE2 | | |
| 00002E | DITOFILED DS CL7 | DOS OUTPUT FILE NAME OVERRIDE | |
| 000035 | DITONUMD DS CL3 | DOS OUTPUT FILE SYSNUM OVERRIDE | |
| 000038 | ORG | | |
| 000038 | DITODTYP DS CL4 | DOS OUTPUT FILE DEVICE TYPE | |
| 00003C | DITFILE3 DS OCL10 | | |
| 00003C | DITSFILE DS CL8 | OS SUSPENSE FILE NAME OVERRIDE | |
| 000044 | DS CL2 | FILLER | |
| 000046 | ORG DITFILE3 | | |
| 00003C | DITSFILED DS CL7 | DOS SUSPENSE FILE NAME OVERRIDE | |
| 000043 | DITSNUMD DS CL3 | DOS SUSPENSE FILE SYSNUM OVERRIDE | |
| 000046 | ORG | | |
| 000046 | DITSDTYP DS CL4 | DOS SUSPENSE FILE DEVICE TYPE | |
| 00004A | DITMXERR DS H | MAX ERROR COUNT FOR INPUT FILE | |
| 00004C | DS XL4 | RESERVED | |
| 00050 | DITLLEN EQU ((*-DIT+3)/4)*4 | LENGTH OF LONG DIT ENTRY | |

1.40 #DNBDS

```
COPY #DNBDS
*****
***          DNB : ADS DIALOG NAME BLOCK
***
***  DNB IS A DSECT THAT DEFINES THE DIALOG NAME BLOCK USED BY
***  THE ADS BATCH GENERATOR. A DNB IS BUILT FOR EACH DIALOG
***  IN THE LOAD AREA WHEN A REQUEST FOR 'ALL' DIALOGS OR A RANGE
***  OF DIALOG NAMES IS INPUT TO THE BATCH GENERATOR. THE LIST
***  OF DNB'S IS MAINTAINED IN SORT SEQUENCE AND IS ANCHORED IN
***  THE GWA.
***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-----------------------|
| 000000 | DNB | DSECT |
| 000000 | DNBNEXTA | DS A |
| 000004 | DNBNAME | DS CL8 |
| 00000C | DNBDVER | DS H |
| 000010 | DNDBKEY | DS F |
| 00014 | DNBLEN | EQU $((*-DNB+3)/4)*4$ |
| 00005 | DNBLENF | EQU $((*-DNB+3)/4)$ |

16:19:45 06/15/82

ADDRESS OF NEXT DNB

DIALOG NAME

DIALOG VERSION

DBKEY OF LOADHDR-156 RECORD

LENGTH OF DNB IN BYTES

LENGTH OF DNB IN WORDS

1.41 #DRNDS

```

COPY #DRNDS
*****
** DSECT FOR DIALOG/APPLICATION REPORT NAME LIST.
**
*****
Offset Value

```

| | | | |
|--------|----------|---------------------|--------------------------------------|
| 000000 | DRN | DSECT | 03/21/88 16:40:05 11/20/90 |
| 000000 | DRNOPTNS | DS H | * REPORT OPTIONS FOR THIS ENTRY |
| 00080 | DRNRPTSU | EQU X'80' | * . OPTION FOR SUMMARY |
| 00040 | DRNRPTPR | EQU X'40' | * . OPTION FOR PROCESSES |
| 00020 | DRNRPTRE | EQU X'20' | * . OPTION FOR RECORDS |
| 00010 | DRNRPTFD | EQU X'10' | * . OPTION FOR FDBLIST |
| 00008 | DRNRPTFS | EQU X'08' | * . OPTION FOR FUNC & RESP-SUMMARY |
| 00004 | DRNRPTFR | EQU X'04' | * . OPTION FOR FUNC * RESP-DETAILD |
| 00002 | DRNRPTXR | EQU X'02' | * . OPTION FOR XREF *LMA86251* |
| 00001 | DRNRPTXS | EQU X'01' | * . OPTION FOR SHORT XREF*90-08-1003 |
| 000002 | DRNFLAG1 | DS X | * MASK OR RANGE SWITCH |
| 000D9 | DRNFLGR | EQU C'R' | * . RANGE FLAG VALUE |
| 000D4 | DRNFLGM | EQU C'M' | * . MASK FLAG VALUE |
| 000003 | DRNFLAG2 | DS X | * RESERVED |
| 000004 | DRNCHAIN | DS A | * CHAIN OF LHR NAMES |
| 000008 | DRNVERS | DS 0F | * PAIR OF VERSIONS FOR NAMES |
| 000008 | DRNVERL | DS H | * LOW VERSION IN RANGE |
| 00000A | DRNVERH | DS H | * HIGH VERSION IN RANGE |
| 00000C | DRNNAMES | DS CL16 | * PAIR OF NAMES OR NAME AND MASK |
| 00001C | ORG | DRNNAMES | |
| 00000C | DRNLOW | DS CL8 | * FOR RANGE, LOW VALUE |
| 000014 | DRNHIGH | DS CL8 | * .. HIGH VALUE |
| 00001C | ORG | DRNNAMES | |
| 00000C | DRNNAME | DS CL8 | * FOR MASK, MATCH NAME |
| 000014 | DRNMASK | DS CL8 | * .. MASK VALUE |
| 00001C | ORG | , | |
| 0001C | DRNLEN | EQU ((*-DRN+3)/4*4) | * LENGTH OF THIS DSECT |

1.42 #DSDWADS

```

#DSDWADS
*****
*          *
*  DSECT FOR ADSOADS WORKING STORAGE (DSDWA)          *
*          *
*****




| <u>Offset</u> | <u>Value</u>                                                     |                |
|---------------|------------------------------------------------------------------|----------------|
| 000000        | DSDWA DSECT                                                      |                |
| 000000        | 0000000000000000SYSPLIST DC 10F'0'                               | PARM LIST AREA |
| 000028        | DS 0D                                                            |                |
| 000028        | SSCTRL DS OCL200                                                 |                |
| 000028        | 4040404040404040PGMNAME DC CL8' '                                |                |
| 000030        | F1F4F0F0ERRSTAT DC C'1400'                                       |                |
| 000034        | 0000000DBKEY DC F'0'                                             |                |
| 000038        | 40404040404040RECNAME DC CL16' '                                 |                |
| 000048        | 40404040404040AREANAME DC CL16' '                                |                |
| 000058        | 40404040404040ERRORSET DC CL16' '                                |                |
| 000068        | 40404040404040ERRORREC DC CL16' '                                |                |
| 000078        | 40404040404040ERRAREA DC CL16' '                                 |                |
| 000088        | SSCIDBCM DS OF                                                   |                |
| 000088        | 0000000000000000IDBMSCOM DC 25F'0'                               |                |
| 0000EC        | 0000000DIRDBKEY DC F'0'                                          |                |
| 0000F0        | DBSTATUS DS OCL7                                                 |                |
| 0000F0        | 4040DBSTMTCDC  CL2' '                                            |                |
| 0000F2        | 404040404040DBSTATCD DC CL5' ',CL1' '                            |                |
| 0000F8        | 0000000RECOCCUR DC F'0'                                          |                |
| 0000FC        | 0000000DMLSEQ DC F'0'                                            |                |
| 000100        | 4040404040404040DSdapnm DC CL8' ' APPLICATION NAME TO BE DELETED |                |
| 000108        | 4040404040404040DSDDICT DC CL8' ' DICTIONARY NAME                |                |
| 000110        | 404040404040404040DSDNODE DC CL8' ' NODE NAME                    |                |
| 000118        | 0001DSDVERS DC H'1' APPLICATION VERSION                          |                |
| 00011A        | 00000SDRTNCD DC H'0' RETURN REASON CODE                          |                |
| 00011C        | 00000000DSDPATCH DC A(0) PATCH AREA ADDRESS                      |                |
| 000120        | DS 0D DBLWORD ALIGN RECORD BUFFER                                |                |
| 000120        | DBBUF DS CL512 SYS-041 RECORD BUFFER                             |                |
| 000320        | DS OF                                                            |                |
| 000320        | ADSCSAA DS F SAVED A(CSA) FOR BATCH/ONLINE *JMA85247*            |                |
| 000324        | ADSIDMSA DS F SAVED A(IDMSIDMS) FOR BATCH *JMA85247*             |                |
| 000328        | ADSENTP DS 3F ENTRY POINT SAVE AREA *JMA85247*                   |                |
| 00334         | DSDWALN EQU *-DSDWA LENGTH OF DSDWA                              |                |


```

1.43 #DWADS

```

COPY #DWADS
*****
***          DWA: ADS/ONLINE GENERATOR PROCESS WORK AREA
***          DWA IS A DSECT THAT DEFINES THE DIALOG NAME BLOCK AREA FOR
***          THE ADS REGENERATOR. THIS KEPT STORAGE IS MANAGED LIKE THE
***          GWA AND PWA BY THE ALCSTG ROUTINE IN ADSGNRC. THIS WORK
***          AREA IS USED TO BUILD THE DNB'S ASSOCIATED WITH REGENERATING
***          MULTIPLE DIALOGS IN THE BATCH REGENERATION ROUTINES.
***          ****
*****
```

Offset Value

| | | | |
|--------|----------|---------------------|------------------------------|
| 000000 | DWA | DSECT | 12:41:24 11/08/82 |
| 000000 | DWAID | DS CL4 | "DWA*" |
| 000004 | DWANXTA | DS A | ADDRESS OF NEXT DWA |
| 000008 | ORG | DWANXTA | |
| | DWAPRIM | #FLAG X'80' | ON IF PRIMARY DWA |
| 000004 | DWAPRIMI | DS OXL1 | |
| 000080 | DWAPRIMM | EQU X'80' | |
| 000004 | | DS X | |
| 000005 | ORG | | |
| 000008 | DWAPREVA | DS A | ADDR OF PREVIOUS DWA |
| 00000C | DWANABA | DS A | ADDR OF NEXT AVAILABLE BYTE |
| 000010 | DWASIZE | DS F | SIZE OF THIS DWA |
| 000014 | DWFRESZ | DS F | AVAILABLE SPACE IN THIS DWA |
| 00018 | DWALEN | EQU ((*-DWA+3)/4)*4 | LENGTH OF DWA HEADER PORTION |

1.44 #DXBDS

```
COPY #DXBDS
*****
*** DXB IS A DSECT THAT DEFINES THE DYNAMIC XDE BLOCK USED BY ***
*** THE ADSO RUNTIME SYSTEM. THIS BLOCK RESIDES IN AN XDE STACK ***
*** AND IS THE SAME SIZE AS AN XDE. DXB'S ARE GENERATED WHEN ***
*** DATA ELEMENTS ARE REFERENCED WHICH ARE SPECIFIC TO A MAP ***
*** FIELD (TYPE 1) AND WHEN TRUE FALSE CONDITIONS ARE ENCOUNTERED ***
*** WHICH ARE EXPRESSED BY A MAP OR DATABASE KEYWORD (TYPE 2). ***
*** A DXB MAY POINT TO AN MRE INDEX LIST WHEN MULTIPLE MAP ***
*** FIELDS ARE REFERENCED AND/OR WHEN IT IS NECESSARY TO OBTAIN ***
*** A SUBSCRIPT VALUE AT RUN TIME. WHEN AN XDE STACK CONTAINS ***
*** ONE OR MORE DXB'S, IT IS MOVED OUT OF THE FDB AND ALL DXB'S ***
*** ARE CONVERTED TO XDE'S BEFORE CALLING RHDEVAL.
*****
```

```
COPY #ADXBDS
*****
*** DXB IS A DSECT THAT DEFINES THE DYNAMIC XDE BLOCK USED BY ***
*** THE ADSO RUNTIME SYSTEM. THIS PORTION OF THE DXB IS ONLY FOR ***
*** USE WITH REDUCED XDE SIZE. SEE THE DXB MACRO FOR A COMPLETE ***
*** EXPLANATION OF THIS CONTROL BLOCK.
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|--|
| 000000 | DXB DSECT | 10:20:48 09/20/84 |
| 000000 | DXBNEXT DS F | OFFSET TO NEXT XDE/DXB |
| 000004 | DXBDTABO DS H | VRE TABLE OFFSET |
| 000006 | DXBDDSPL DS H | DATA OFFSET (OR ZEROS) |
| 000008 | DXBDYN DS H | VXDE OFFSET |
| * | | |
| | ** FOLLOWING FLAGS APPLY TO TYPE 2 DXB'S FOR DB-STATUS TESTS | |
| | ** FLAG 1 DXBTF BIT ON AND DXBMAP BIT OFF | |
| * | | |
| 00000A | DXBDBST #FLAG X'80' | ON MEANS TEST DB-STATUS-OK |
| 000080 | DXBDBSTI DS 0XL1 | |
| | DXBDBSTM EQU X'80' | |
| 00000A | DXBDBES #FLAG X'40' | ON MEANS TEST DB-END-OF-SET |
| 000040 | DXBDBESI DS 0XL1 | |
| | DXBDBESM EQU X'40' | |
| 00000A | DXBDBBRN #FLAG X'20' | ON MEANS TEST DB-REC-NOT-FOUND |
| 000020 | DXBDBRNI DS 0XL1 | |
| | DXBDBRNM EQU X'20' | |
| 00000A | DXBSETE #FLAG X'10' | ON MEANS TEST DB-SET-EMPTY |
| 000010 | DXBSETEI DS 0XL1 | |
| | DXBSETEM EQU X'10' | |
| 00000A | DXBSETM #FLAG X'08' | ON MEANS TEST DB-SET-MEMBER |
| 000008 | DXBSETMI DS 0XL1 | |
| | DXBSETMM EQU X'08' | |
| * | X'00' | ALL BITS OFF MEANS TEST DB-ANY-ERROR |
| * | | |
| | ** FOLLOWING FLAGS APPLY TO TYPE 2 DXB'S FOR MAP CONDITION TESTS | |
| | ** FLAG 1 DXBTF BIT ON AND DXBMAP BIT ON | |
| * | | |
| 00000A | DXBALLM #FLAG X'80' | ON MEANS TEST APPLIES TO ALL MAP FIELDS |
| 000080 | DXBALLMI DS 0XL1 | |
| | DXBALLMM EQU X'80' | |
| 00000A | DXBABUT #FLAG X'48' | ON MEANS ALL BUT LISTED FIELDS MUST PASS |
| 000048 | DXBABUTI DS 0XL1 | |
| | DXBABUTM EQU X'48' | |
| 00000A | DXBALOF #FLAG X'40' | ON MEANS ALL FIELDS MUST PASS THE TEST |
| 000040 | DXBALOFI DS 0XL1 | |
| | DXBALOFM EQU X'40' | |

| | | | |
|--------|----------|---------------------|---|
| 00000A | DXBANY | #FLAG X'20' | ON MEANS ANY FIELD MUST PASS THE TEST |
| | 00020 | DXBANYI DS OXL1 | |
| | | DXBANYM EQU X'20' | ON MEANS NO FIELD MAY PASS THE TEST |
| 00000A | DXBNONE | #FLAG X'10' | |
| | 00010 | DXBNONEI DS OXL1 | |
| | | DXBNONEM EQU X'10' | ON MEANS SOME FIELDS MUST PASS THE TEST |
| 00000A | DXBSOME | #FLAG X'08' | |
| | 00008 | DXBSOMEI DS OXL1 | |
| | | DXBSOME M EQU X'08' | ON MEANS "IDENTICAL" TEST SPECIFI*JMA86065* |
| 00000A | DXBIDEN | #FLAG X'05' | |
| | 00005 | DXBIDENI DS OXL1 | |
| | | DXBIDENM EQU X'05' | ON MEANS "CHANGED" TEST SPECIFIED |
| 00000A | DXBCHNG | #FLAG X'04' | |
| | 00004 | DXBCHNGI DS OXL1 | |
| | | DXBCHNGM EQU X'04' | ON MEANS "IN ERROR" TEST SPECIFIED |
| 00000A | DXBERRS | #FLAG X'03' | |
| | 00003 | DXBERRSI DS OXL1 | |
| | | DXBERRSM EQU X'03' | ON MEANS "TRUNCATED" TEST SPECIFIED |
| 00000A | DXBTRUN | #FLAG X'02' | |
| | 00002 | DXBTRUNI DS OXL1 | |
| | | DXBTRUNM EQU X'02' | ON MEANS "ERASED" TEST SPECIFIED |
| 00000A | DXBERASI | DS OXL1 | |
| | 00001 | DXBERASM EQU X'01' | |
| | * | X'00' | ALL BITS OFF MEANS "CURSOR AT FLD-NAME" |
| 00000A | DXBFLG2 | DS X | DXB FLAG BYTE 2 |
| 00000B | DXBDATLN | DS X | DATA LENGTH |
| | | DXBID #FLAG X'80' | ALWAYS ON, IDENTIFIES A DXB (NOT AN XDE) |
| 00000C | DXBIDI | DS OXL1 | |
| | 00080 | DXBIDM EQU X'80' | |
| | | DXBTF #FLAG X'40' | ON IF A TRUE/FALSE (TYPE 2) DXB |
| 00000C | DXBTFI | DS OXL1 | |
| | 00040 | DXBTFM EQU X'40' | |
| | | DXBMIL #FLAG X'20' | ON IF DXBINDEX IS LITPOOL/PROCESS OFFSET |
| 00000C | DXBMILI | DS OXL1 | |
| | 00020 | DXBMILM EQU X'20' | |
| | | DXBLITP #FLAG X'10' | ON IF DXBINDEX OFFSET IS INTO THE LITPOOL |
| 00000C | DXBLITPI | DS OXL1 | |
| | 00010 | DXBLITPM EQU X'10' | |
| | | DXBNOT #FLAG X'08' | ON IF NOT CONDITION IS TRUE (TYPE 2 ONLY) |
| 00000C | DXBNOTI | DS OXL1 | |
| | 00008 | DXBNOTM EQU X'08' | |
| | | DXBMAP #FLAG X'04' | ON IF MAP CONDITION TEST (TYPE 2 ONLY) |
| 00000C | DXBMAPI | DS OXL1 | |
| | 00004 | DXBMAPM EQU X'04' | |
| | | DXBSGFD #FLAG X'02' | ON IF SINGLE MAPFIELD TEST (TYPE 2 ONLY) |
| 00000C | DXBSGFDI | DS OXL1 | |
| | 00002 | DXBSGFDM EQU X'02' | |
| | | DXBAB2 #FLAG X'01' | ON IF 2ND PASS THRU ALL BUT ECM87293 |
| 00000C | DXBAB2I | DS OXL1 | |
| | 00001 | DXBAB2M EQU X'01' | |
| 00000C | DXBFLG1 | DS X | DXB FLAG BYTE 1 |
| 00000D | DXBDATYP | DS X | DATA TYPE USED BY ADSO GENERATOR |
| 00000E | DXBINDEX | DS H | RDE OR MRE INDEX OR OFFSET TO MRE INDEX |
| | * | | LIST OR TO SET NAME (SET-MEMBER/EMPTY) |
| 00010 | DXBLEN | EQU *-DXB | LENGTH OF DXB |

1.45 #EFHDRDS

COPY #EFHDRDS

| <u>Offset</u> | <u>Value</u> | |
|---------------|------------------------|--|
| 000000 | EFHDRDS DSECT | 12:01:34 05/18/84 |
| | * | EVAL FUNCTION MODEL TABLE HEADER DSECT |
| 000000 | EFHNEXT DS H | OFFSET TO NEXT HDR ENTRY |
| 000002 | EFHUNNM DS CL8 | FUNCTION NAME - REAL |
| 00000A | EFHPPGMN DS CL8 | PROCESSING PROGRAM NAME |
| 000012 | EFHPPGMV DS H | PROCESSING PROGRAM VERSION |
| 000014 | EFHFUNCN DS XL1 | FUNCTION NUMBER |
| 000015 | DS XL1 | FILLER |
| 000016 | EFHWORKL DS H | LENGTH OF REQUIRED WORKAREA |
| 000018 | EFHZOPND DS 0XL4 | 4 X'00'S INDICATE ZERO OPERANDS |
| 000018 | EFHFOPDN DS H | NUMBER OF FIXED OPERANDS |
| 00001A | EFHVOPDO DS H | OFFSET TO VARIABLE OPERAND MODEL |
| 00001C | EFHRESLN DS H | RESULT LENGTH IN BYTES |
| 00001E | EFHRDATP DS XL1 | RESULT DATA TYPE |
| 00001F | EFHRNDEC DS XL1 | RESULT NUMBER DECIMALS |
| 000020 | DS XL4 | FILLER |
| 00024 | EFHDRLNG EQU *-EFHDRDS | LENGTH OF FUNCTION MODEL HEADER |

1.46 #EFMASDS

COPY #EFMASDS

| <u>Offset</u> | <u>Value</u> | |
|---------------|------------------------|--|
| 000000 | EFMASDS DSECT | 11:15:30 03/06/86 |
| | * | EVAL MASTER FUNCTION TABLE ENTRY DSECT |
| 000000 | EFMINAML DS H | LENGTH OF INVOCATION FUNCT NAME |
| 000002 | EFMINAME DS CL32 | FUNCTION NAME - INVOCATION |
| 000022 | EFMRNAME DS CL8 | FUNCTION NAME - REAL |
| 00002A | EFMMPGMN DS CL8 | PROGRAM NAME - MODEL XDE TABLE |
| 000032 | EFMMPGMV DS H | PROGRAM VERSION - MODEL XDE TABLE |
| 000034 | EFMFLAG1 DS XL1 | MASTER FUNCTION ENTRY FLAG1 |
| 000080 | EFMASFU EQU X'80' | AGGREGATE FUNCTION ENTRY |
| 000035 | DS XL3 | FILLER |
| 00038 | EFMASLNG EQU *-EFMASDS | LENGTH OF MASTER ENTRY |

1.47 #EFXDEDS

COPY #EFXDEDS

| <u>Offset</u> | <u>Value</u> | |
|---------------|-----------------------|--------------------------------------|
| 000000 | EFXDEDS DSECT | 07:36:43 05/31/84 |
| | * | EVAL FUNCTION MODEL XDE DSECT |
| 000000 | EFXNEXT DS H | OFFSET TO NEXT MODEL XDE |
| 000002 | EFXNDEC DS XL1 | NUMBER OF DECIMALS |
| 000003 | EFXRLCF DS XL1 | RESULT LENGTH CALCULATION FLAG |
| 000080 | EFXRSCP EQU X'80' | ADD LENGTH |
| 000040 | EFXRCLS EQU X'40' | SUBT LENGTH |
| | * | IF ZERO, IGNORE |
| 000004 | EFXFLAG1 DS XL1 | FIRST FLAG |
| 000080 | EFXF1MAN EQU X'80' | ON=MANDATORY, OFF=OPTIONAL |
| 000040 | EFXF1TRU EQU X'40' | ON=TRUNCATE, OFF=ROUND |
| 000020 | EFXF1RES EQU X'20' | RESULT CHARACTERISTICS DEFAULT |
| 000005 | DS XL3 | FILLER |
| 000008 | EFXDCTN DS H | NUMBER OF ENTRIES IN DATA CONV TBL |
| 00000A | EFXLNG1 EQU *-EFXDEDS | BASE LENGTH OF ENTRY |
| | * | DATA TYPE CONVERSION TABLE |
| 000000 | EFXCNVE DSECT | CONVERSION TBL ENTRY DSECT*LCB84143* |
| 000000 | EFXSRCT DS XL1 | SOURCE DATATYPE |
| 000001 | EFXTART DS XL1 | TARGET DATATYPE |
| 000002 | EFXTARL DS H | TARGET LENGTH |
| 000004 | DS XL2 | FILLER |
| 000006 | EFXDCTL EQU *-EFXSRCT | LENGTH OF ENTRY |

1.48 #EXCDS

```

COPY #EXCDS
*****
***   EXC IS THE VECTOR PROTOTYPE DSECT USED FOR BUILDING CODE FOR ***
***   THE EXECUTABLE CME'S WITHIN THE FDB.                               ***
***   *****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|------------------------------------|
| 000000 | EXC DSECT | 09:01:54 03/19/84 |
| 000000 | EXCNEXTA DS F | ADDR OF NEXT EXC IN CHAIN |
| 000004 | EXCPREVA DS F | ADDR OF PRIOR EXC IN CHAIN |
| 000008 | EXCSIZE DS H | LEN OF EXEC CODE FOLLOWING EXCDATA |
| 00000A | EXCPRMSZ DS H | LEN OF PARMS (CME FOR VECTOR CALL) |
| 00000C | EXCPRMOF DS F | OFFSET OF PARMS |
| | * Prior to Release R11.0, the following two lengths were computed | |
| | * during code generation but never used subsequently. | |
| | * For R11.0, we no longer compute these values. | |
| 000010 | EXCACLEN DS F | ACCUMULATE LENGTH OF EXECUTABLE CD |
| 000014 | EXCPRALN DS F | ACCUM LEN OF PARAMETERS FOR CODE |
| | * | |
| 000018 | EXCFUNC DS H | VECTOR CALL FUNCTION CODE |
| | EXCCOND #FLAG X'80' | SIMPLE CONDITIONAL |
| 00001A | EXCCONDI DS OXL1 | |
| | 000080 EXCCOND# EQU X'80' | |
| | EXCBRCH #FLAG X'40' | Code is for a Branch CME |
| 00001A | EXCBRCHI DS OXL1 | |
| | 000040 EXCBRCHM EQU X'40' | |
| 00001A | EXCFLAG1 DS X | 1ST FLAG BYTE |
| 00001B | DS X | RESERVED |
| 00001C | EXCTFOF DS H | OFFSET TO TRUE/FLSE PAIR |
| 00001E | DS H | RESERVED |
| 000020 | DS F | RESERVED |
| | 000024 EXCLEN EQU ((*-EXC)/4)*4 | LENGTH OF EXC FIXED PORTION |
| 000009 | EXCLENF EQU ((*-EXC+3)/4) | # OF FULLWORDS IN FIXED PORTION |
| 000024 | EXCDATA DS 0F | |

1.49 #EXCFNDS

```
COPY #EXCFNDS
*****
*      EXECUTABLE CODE FUNCTION TABLE
*
*      THE FIRST 2 HALFWORDS IN EACH ENTRY OF THIS TABLE ARE THE
*      CME CLASS CODE AND THE CME FUNCTION CODE.  THE THIRD VALUE
*      IS THE VECTOR FUNCTION CODE TO BE USED IN EXECUTABLE CODE
*      PROCESSING IN ORDER TO ROUTE CME'S TO THE APPROPRIATE RUN-
*      TIME ROUTINE.
*
*****
Offset  Value
000000    EXCFNDS  DSECT          11:49:33 09/07/84
000000    EXCFNCL  DS   H          CME CLASS
000002    EXCFNFn  DS   H          CME FUNCTION
000004    EXCFNVC  DS   H          CME VECTOR
000006    EXCFNLEN EQU  *-EXCFNDS
```

1.50 #EXEDS

```

COPY #EXEDS
*****
***          EXE: EXPRESSION ENTRY DSECT
***          ***
***          EXE IS A DSECT THAT DESCRIBES THE ENTRIES IN THE
***          POSTFIX EXPRESSION CREATED BY THE ADSO BATCH GENERATOR.
***          EACH ENTRY IN THE POSTFIX EXPRESSION REPRESENTS EITHER
***          AN OPERATOR OR AN OPERAND. AN OPERAND MAY BE A
***          CONSTANT, A VARIABLE, OR A SUBSCRIPTED VARIABLE. A
***          SUBSCRIPTED VARIABLE REQUIRES A DOPE VECTOR WHICH
***          CONTAINS INFORMATION ABOUT THE SIZE OF EACH DIMENSION.
***          ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-------------------------|-----------------------------------|
| 000000 | EXE DSECT | 12:05:15 08/13/80 |
| | EXEOPND #FLAG X'80' | SET FOR OPERAND ENTRY |
| 000000 | EXEOPNDI DS 0XL1 | |
| 000080 | EXEOPNDM EQU X'80' | |
| 000000 | EXEENTYP DS X | ENTRY FLAG |
| | EXESUBO #FLAG X'80' | SET FOR SUBSCRIPTED OPERAND |
| 000001 | EXESUBOI DS 0XL1 | |
| 000080 | EXESUBOM EQU X'80' | |
| | EXEDVIN #FLAG X'40' | SET FOR INLINE DOPE VECTOR |
| 000001 | EXEDVINI DS 0XL1 | |
| 000040 | EXEDVINM EQU X'40' | |
| 000001 | EXEEFLG DS X | FLAG BYTE |
| 000002 | EXEOPTYP DS H | OPERATOR TYPE |
| 000004 | ENUMOP DS H | NUMBER OF OPERANDS EXECUTED |
| 000006 | EXEOPRLN EQU *-EXEENTYP | SIZE OF OPERATOR ENTRY |
| 000006 | ORG EXEOPTYP | |
| 000002 | EXEDATYP DS H | DATA TYPE |
| 000004 | EXEDECPL DS H | NUMBER OF DECIMAL PLACES |
| 000006 | EXERECIX DS H | RECORD INDEX IN VPB |
| 000008 | EXEITMIX DS H | ITEM OFFSET IN REC/GROUP |
| 00000A | EXEITMSZ DS H | SIZE OF ITEM |
| 00000C | EXEOPNLN EQU *-EXEENTYP | SIZE OF OPERAND ENTRY |
| 00000C | EXESSNUM DS H | NUMBER OF SUBSCRIPTS |
| 00000E | DS H | RESERVED |
| 000010 | EXEDOPEA DS A | ADDRESS OF DOPE VECTOR |
| 00014 | EXESOPLN EQU *-EXEENTYP | SIZE OF SUBSCRIPTED OPERAND ENTRY |

1.51 #FDBDS

```

COPY #FDBDS
*****
*** FDB IS A DSECT THAT DEFINES THE FIXED DIALOG BLOCK USED BY ***
*** THE ADSO RUNTIME SYSTEM. THIS BLOCK CONTAINS ALL THE ***
*** INFORMATION NEEDED TO EXECUTE AN ADSO DIALOG. THERE WILL ***
*** BE ONE FDB ACTIVE AT ANY GIVEN TIME FOR A PARTICULAR ***
*** DIALOG. ANY ADSO TASK USING THE DIALOG WILL REFERENCE ***
*** THE COMMON FDB FOR THE DIALOG AND ALSO CREATE A VARIABLE ***
*** DIALOG BLOCK (VDB) USED BY THE INDIVIDUAL TASK TO CONTAIN ***
*** THE VARIABLE INFORMATION NEEDED FOR EACH EXECUTION OF THE ***
*** DIALOG. ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|---|-------------|--|
| 000000 | FDB | DSECT | 03/09/88 14:22:48 12/07/90 |
| 000000 | FDBID | DS CL4 | "FDB*" |
| 000004 | FDBNAME | DS CL8 | DIALOG NAME |
| 00000C | FDBDATE | DS CL8 | DIALOG GEN DATE |
| 000014 | FDBTIME | DS CL8 | DIALOG GEN TIME |
| 00001C | FDBMPNM | DS CL8 | MAP NAME |
| 000024 | FDBMPDT | DS CL8 | DATE MAP COMPILED |
| 00002C | FDBMPTM | DS CL8 | TIME MAP COMPILED |
| 000034 | FDBSCHNM | DS CL8 | SCHEMA NAME |
| 00003C | FDBSSNM | DS CL8 | SUBSCHEMA NAME |
| 000044 | FDBRDEA | DS F | OFFSET OF START OF RECORD TBL |
| 000048 | FDBPMEA | DS F | OFFSET OF START OF PREMAP ELEMNT |
| 00004C | FDBRSEA | DS F | OFFSET OF START OF RESPONSE TBL |
| 000050 | FDBLITA | DS F | OFFSET OF START OF LITERAL POOL |
| 000054 | FDBSSANA | DS F | OFFSET OF SUBS AREA NAME TABLE |
| 000058 | FDBNSSAN | DS H | # OF SUBSCHEMA AREA NAMES |
| 00005A | FDBSVER | DS H | SCHEMA VERSION |
| 00005C | FDBMPVER | DS H | MAP VERSION |
| 00005E | FDBDVER | DS H | DIALOG VERSION |
| 000060 | FDBNRECS | DS H | # OF MAP RECORDS |
| 000062 | FDBNFLDS | DS H | # OF MAP FIELDS |
| 000064 | FDBNDREC | DS H | # OF DIALOG RECORDS |
| 000066 | FDBRSPMI | DS H | MRE INDEX OF RESPONSE FIELD |
| 000068 | FDBMSGMI | DS H | MRE INDEX OF MESSAGE FIELD |
| 00006A | FDBSEGVIEW | DS X | MRB - SUBSCHEMA SEGMENTED VIEW |
| | | PRINT NOGEN | |
| | FDBMAIN | #FLAG X'80' | ON IF MAINLINE DIALOG |
| | FDBNOMP | #FLAG X'40' | ON IF NO MAP FOR DIALOG |
| | FDBAUTO | #FLAG X'20' | ON IF AUTOSTATUS BEING USED |
| | FDBEXTM | #FLAG X'10' | ON IF EXTENDED MRB REQUIRED |
| | FDBRSPF | #FLAG X'08' | ON IF MAP HAS A RESPONSE FIELD |
| | FDBMSGF | #FLAG X'04' | ON IF MAP HAS A MESSAGE FIELD |
| | FDBEXCD | #FLAG X'02' | ON IF EXECUTABLE CODE USED |
| | FDBACT | #FLAG X'01' | ON IF ACTIVITY LOGGING ENABLED |
| 00006B | FDBFLAG | DS X | FDB FLAG BYTE |
| 00006C | FDBLREA | DS A | OFFSET OF FIRST RDE FOR LR |
| 000070 | FDBASRA | DS F | OFFSET TO AUTOSTATUS REC ASR |
| | | ***** | |
| | * FDBRLSE CONTAINS A CODE WHICH REPRESENTS THE RELEASE LEVEL *MCM86220* | | |
| | * WHICH WAS USED TO GENERATE THE DIALOG. THE ENCODING IS AS *MCM86220* | | |
| | * FOLLOWS : *MCM86220* | | |
| | * *MCM86220* | | |
| * | RELEASE | 1.0 | ====> X'0000' (FIELD NOT USED FOR 1.0)*MCM86220* |
| * | RELEASE | 1.1 | ====> X'1100' (ONLY 1 BYTE FOR 1.1) *MCM86220* |
| * | RELEASE | 10.0 | ====> X'0100' (MADE 2 BYTES FOR 10.0) *MCM86220* |
| * | RELEASE | 10.1 | ====> X'0101' *MCM86220* |
| * | RELEASE | 10.2 | ====> X'0102' *MCM87341* |

```

* ADS/CICS Release 1.0 ===> X'0103' *MCM87341*
* Release 10.2.1 ===> X'0104' *MCM87341*
* Release 11.0 ===> X'0110' *MCM87341*
* RELEASE 12.0 ===> X'0120' *MET90341*
* RELEASE WXYZ ===> X'WXYZ' (GENERAL FORMAT) *MCM86220*
*
* NOTE THAT DUE TO THE CHANGE FROM 1 TO 2 BYTES, RELEASE *MCM86220*
* 110.0 WILL HAVE THE SAME ENCODING AS RELEASE 1.1 (X'1100') *MCM86220*
*****
```

| | | |
|--------|---|---|
| 000074 | FDBRLSE DS AL2 | ADS RELEASE LEVEL |
| | FDBSYMT #FLAG X'80' | ON IF SYMBOL TABLE IN USE |
| | FDBCOBM #FLAG X'40' | ON IF COBOL MOVES IN EFFECT |
| | FDBNBKP #FLAG X'20' | ON IF NO BACKPAG OPTION IN USE |
| | FDBBRWS #FLAG X'10' | ON IF BROWSE (VS. UPDATE) MODE |
| | FDBENTP #FLAG X'08' | ON IF MAP ENTRY POINT *JMA85077* |
| | FDBPAGM #FLAG X'04' | ON IF DIALOG HAS PAGEABLE MAP |
| | FDB\$PAG #FLAG X'02' | ON IF MAP HAS A \$PAGE FIELD |
| | FDBNTBL #FLAG X'01' | NO LNT,OFT,MDT TBLS GE*LC884199* |
| 000076 | FDBFLAG2 DS X | FDB 2ND FLAG BYTE |
| | PRINT GEN | |
| | *** FOLLOWING EQUATES SPECIFY LEGAL VALUES FOR FDBMAPPG | |
| 000E6 | FDBWAIT EQU C'W' | WAIT MODE |
| 000D5 | FDBNOWT EQU C'N' | NOWAIT MODE |
| 000D9 | FDBRETN EQU C'R' | RETURN MODE |
| | * | |
| 000077 | FDBMAPPG DS X | MAP PAGING FLAG |
| 000078 | FDBHEXTA DS F | OFFSET TO FDB HEADER EXTENSION |
| 00007C | FDBMDBO DS F | OFFSET TO MAP DESCRIPTOR BLOCK |
| | FDBBIM #FLAG X'80' | ON IF BATCH INPUT MAP PRESENT |
| 000080 | FDBBIMI DS OXL1 | |
| 000080 | FDBBIMM EQU X'80' | |
| | FDBBOM #FLAG X'40' | ON IF BATCH OUTPUT MAP PRESENT |
| 000080 | FDBBOMI DS OXL1 | |
| 000040 | FDBBOMM EQU X'40' | |
| | FDBNRLK #FLAG X'20' | ON IF RETRIEVAL LOCKS NOT KEPT |
| 000080 | FDBNRLKI DS OXL1 | |
| 000020 | FDBNRLKM EQU X'20' | |
| | FDBNADS #FLAG X'10' | ON IF NO MP AUTO DSPLY DKJ86211 |
| 000080 | FDBNADSI DS OXL1 | |
| 000010 | FDBNADSM EQU X'10' | |
| 000080 | FDBFLAG3 DS X | FDB 3RD FLAG BYTE |
| 000081 | FDBPREFIX DS CL2 | MESSAGE PREFIX *LMA86162* |
| 000083 | DS X | RESERVED *LMA86162* |
| 000084 | FDBDRSP0 DS F | OFFSET TO DEFAULT RESPONSE |
| 000088 | FDBFDE0 DS F | Offset to start of FDE*MCM88069 area. *MCM88069 |
| | * | |
| 0008C | FDBLEN EQU *-FDB | LENGTH OF FDB FIXED PORTION |

1.52 #FDEDS

```

COPY #FDEDS
*****
*** FDE IS THE DSECT THAT DEFINES THE FUNCTION DEFINITION ***
*** ELEMENT WHICH REPRESENTS A FUNCTION WITHIN AN ADS ***
*** APPLICATION. FDE LENGTHS ARE SPECIFIED IN THE ADB HEADER. ***
*** ALL FDE'S IN A GIVEN ADB ARE THE SAME LENGTH SO THAT THEY ***
*** MAY BE LOCATED FROM THE RESPONSE APRE'S VIA AN INDEX. THE ***
*** CONTENT OF AN FDE VARIES STARTING AT FIELD FDEULFE DEPEND- ***
*** ING ON THE TYPE OF FUNCTION. FDE'S WHICH INVOKE PROGRAMS ***
*** HAVE A LITPOOL FINDER ELEMENT (LFE) TO FIND THE PROGRAM'S ***
*** USING RECORDS LIST. IN THE PLACE OF THE LFE, FDE'S WHICH ***
*** INVOKE MENU FUNCTIONS HAVE AN INDEX INTO THE MENU EXTENSION ***
*** TABLE (MET) TO THE MENU TEXT ELEMENT (MTE). THE LAST ELE- ***
*** MENTS OF AN FDE ARE TWO BITMAPS. THE FIRST LOCATES APRE'S ***
*** (APPLICATION RESPONSE ELEMENTS) IN THE APRE TABLE WHICH ***
*** REPRESENT VALID RESPONSES FOR THE FUNCTION. THE SECOND ***
*** BITMAP LOCATES RESPONSE KEY ELEMENTS (RKE) FOR TERMINAL ***
*** KEYBOARD KEYS WHICH ARE SYNONYMOUS TO VALID RESPONSES. NOTE ***
*** THAT FOR APPLICATIONS WHICH HAVE NO DEFINED VALID KEYS, ***
*** THE LENGTH OF THE RKE BITMAP MAY BE ZERO. FUNCTIONS WHICH ***
*** INVOKE DC PROGRAMS HAVE NO VALID RESPONSES AND, THEREFORE ***
*** THE CONTENT OF THEIR BITMAPS IS IRRELEVANT.
*** ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|---|
| 000000 | FDE DSECT | 12:15:05 04/04/85 |
| 000000 | FDEID DS CL8 | FUNCTION NAME |
| 000008 | FDENAME DS CL8 | NAME OF DIALOG OR DC PROGRAM INVOKED |
| 000010 | FDEUXITD DS CL8 | USER EXIT DIALOG NAME |
| 000018 | FDERBMOF DS F | BITMAP POOL OFFSET TO VALID RESPONSES |
| 00001C | FDEKBMOF DS F | BITMAP POOL OFFSET TO FUNCTION KEY RSPS |
| 000020 | FDEINDX DS H | INDEX OF THIS FDE RELATIVE TO ZERO |
| 000022 | FDEDRIX DS H | INDEX OF DEFAULT RESPONSE APRE |
| 000024 | FDEDLFE DS F | DESCRIPTION LIT POOL FINDER |
| 000028 | ORG FDEDLFE | |
| 000024 | FDEDLEN DS X | FUNCTION DESCRIPTION LENGTH |
| 000025 | FDEDLOF DS XL3 | LIT POOL OFFSET TO FUNCTION DESCRIPTION |
| 000028 | ORG | |
| | PRINT NOGEN | |
| | FDEDIAL #FLAG X'80' | ON = FUNCTION INVOKES A DIALOG |
| | FDEPROG #FLAG X'40' | ON = FUNCTION INVOKES A DC PROGRAM |
| | FDEMENU #FLAG X'20' | ON = FUNCTION IS A MENU FUNCTION |
| | * NOTE: THE FOLLOWING FLAGS ARE USED ONLY BY ADSA GENERATOR *MDR* | |
| | FDEUNDF #FLAG X'10' | ON = FUNCTION IS UNDEFINED |
| | FDEGBAD #FLAG X'08' | ON = GLOBAL RESPONSE ADD |
| 000028 | FDEFLAG DS X | FDE FLAG BYTE |
| | PRINT GEN | |
| 000029 | DS X | RESERVED |
| 00002A | DS H | RESERVED |
| 00002C | DS 3F | RESERVED |
| | * | |
| | ** FOLLOWING FIELDS APPLY ONLY TO FUNCTIONS WHICH INVOKE PROGRAMS | |
| 000038 | FDEULFE DS F | PGM'S USING LIST LITPOOL FINDER |
| 00003C | ORG FDEULFE | |
| 000038 | FDEULEN DS H | USING LIST LENGTH |
| 00003C | FDEULOF DS F | LITPOOL OFFSET TO USING LIST |
| | * | |
| | ** FOLLOWING FIELD APPLIES ONLY TO MENU FUNCTIONS | |
| 000040 | ORG FDEULFE | |

| | | | | |
|--------|-----------|-------|---------------|-------------------------------------|
| 000038 | FDEMTEO | DS | H | MENU EXTENSION TABLE OFFSET TO MTE |
| 00003A | FDENMTES | DS | H | NUMBER OF MENU HDR LINES (MTE'S) |
| 00003C | FDERPP | DS | X | NUMBER OF RESPONSES PER PAGE |
| | FDEMSGN | #FLAG | X'80' | ON IF SIGNON TYPE MENU |
| 00003D | FDEMSGNI | DS | 0XL1 | |
| 000080 | FDEMSGNM | EQU | X'80' | |
| | FDEMDSL | #FLAG | X'40' | ON IF LONG DESCRIPTION FOR FUNCTION |
| 00003D | FDEMDSL1 | DS | 0XL1 | |
| 000040 | FDEMDSL1M | EQU | X'40' | |
| | FDEMRSQ | #FLAG | X'20' | ON IF RESPONSE SEQUENCING REQUIRED |
| 00003D | FDEMRSQI | DS | 0XL1 | |
| 000020 | FDEMRSQM | EQU | X'20' | |
| 00003D | FDEMNUL | DS | X | MENU FLAG BYTE |
| 00003E | | ORG | | |
| 000040 | FDELEN | EQU | *-FDE | LENGTH OF FIXED FDE |
| 00010 | FDELENF | EQU | ((*-FDE+3)/4) | LENGTH OF FIXED FDE (WORDS) |

1.53 #FHEDS

```

COPY #FHEDS
*****
*          FHE: FDB HEADER EXTENSION
*****
Offset Value

000000      FHE      DSECT           04/19/88 09:44:27  04/30/91
000000      FHENODE  DS    CL8        ALTERNATE DB NAME
000008      FHEDICT  DS    CL8        ALTERNATE DICTIONARY NAME
*          THE SUSPENSE FILE DDNAME WILL BE 8 BYTES FOR OS AND
*          7 BYTES FOR DOS (LAST BYTE WILL BE IGNORED).
000010      FHESDDN DS    CL8        SUSPENSE FILE DD NAME
000018      FHEDCLA DS    F          OFFSET OF START OF SQL DECLARATION
00001C      FHESQLAM DS    CL8        SQL ACCESS MODULE NAME      MET90341
00001C      FHESQTU  #FLAG X'80'   USA    SQL TIME STANDARD     JMA91119
000024      FHESQTUI DS    0XL1
000024      00080   FHESQTUM EQU  X'80'
000024      FHESQTII DS    0XL1
000024      00040   FHESQTIM EQU  X'40'
000024      FHESQTE  #FLAG X'20'   EUR    SQL TIME STANDARD     JMA91119
000024      FHESQTEI DS    0XL1
000024      00020   FHESQTEM EQU  X'20'
000024      FHESQTJ  #FLAG X'10'   JIS    SQL TIME STANDARD     JMA91119
000024      FHESQTJI DS    0XL1
000024      00010   FHESQTJM EQU  X'10'
000024      FHESQLTM DS    X          SQL TIME FORMAT         MET90341
000024      FHESQDU  #FLAG X'80'   USA    SQL DATE STANDARD     JMA91119
000025      FHESQDUI DS    0XL1
000025      00080   FHESQDUM EQU  X'80'
000025      FHESQDI  #FLAG X'40'   ISO    SQL DATE STANDARD     JMA91119
000025      FHESQDII DS    0XL1
000025      00040   FHESQDIM EQU  X'40'
000025      FHESQDE  #FLAG X'20'   EUR    SQL DATE STANDARD     JMA91119
000025      FHESQDEI DS    0XL1
000025      00020   FHESQDEM EQU  X'20'
000025      FHESQDJ  #FLAG X'10'   JIS    SQL DATE STANDARD     JMA91119
000025      FHESQDJI DS    0XL1
000025      00010   FHESQDJM EQU  X'10'
000025      FHESQDLT DS    X          SQL DATE FORMAT        MET90341
000025      FHESQAN  #FLAG X'80'   ANSI   STD SQL SYNTAX CHECK  JMA91092
000026      FHESQANI DS    0XL1
000026      00080   FHESQANM EQU  X'80'
000026      FHESQFP  #FLAG X'40'   FIPS   STD SQL SYNTAX CHECK  JMA91119
000026      FHESQFPI DS    0XL1
000026      00040   FHESQFPM EQU  X'40'
000026      FHESQDB  #FLAG X'20'   DB2    STD SQL SYNTAX CHECK  JMA91119
000026      FHESQDBI DS    0XL1
000026      00020   FHESQDBM EQU  X'20'
000026      FHESQS2  #FLAG X'10'   SQL2   STD SQL SYNTAX CHECK  JMA91119
000026      FHESQS2I DS    0XL1
000026      00010   FHESQS2M EQU  X'10'
000026      FHESQSA  #FLAG X'08'   SAA    STD SQL SYNTAX CHECK  JMA91119
000026      FHESQSAI DS    0XL1
000026      00008   FHESQSAM EQU  X'08'
000026      FHESQLFL DS    X          SQL COMPLIANCE FLAG      MET90341
000027      DS      XL9        RESERVED             JMA91092
*****
*          THE ONLY MODULE SENSITIVE TO THE FHELEN VALUE IS ADSOGFDB. *MET90341
*          IF MORE SPACE IS NEEDED FOR FDB HEADER INFORMATION, THE      *MET90341
*          FOLLOWING IS RECOMMENDED. ADD SPACE TO THE FHE AND      *MET90341
*          CREATE A NEW LENGTH SYMBOLIC WITH A NAME LIKE FHELEN2.      *MET90341

```

```
* REVISE GFDB TO USE THIS NEW LENGTH. ANY MODULES THAT      *MET90341
* REFERENCE FIELDS IN THIS NEW EXTENSION WILL HAVE TO FIRST *MET90341
* TEST THE FDBRLSE FIELD TO DETERMINE IF THIS AREA EXISTS,   *MET90341
* SO FDBRLSE WILL HAVE TO BE CHANGED. SEE THE FDB MACRO      *MET90341
* FOR POSSIBLE FDBRLSE VALUES.                                *MET90341
*****MET90341
00030 FHELEN EQU *-FHE
```

1.54 #FMDDS

```

COPY #FMDDS
*****
*          ---FILE MAPPING DIRECT-CALL TABLE DSECT---
*
* THIS DSECT MAPS THE TABLE IN RHDCBCTL THAT CONTAINS
* ENTRIES FOR ALL COMBINATIONS OF OPERATING SYSTEM,
* DATASET ORGANIZATION AND DATASET USAGE THAT ARE VALID (SUPPORTED)
* BY FILE MAPPING.
*
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|------------------------------|---------------------------------|
| 000000 | FMDDS DSECT | 11:20:33 06/11/86 |
| 000000 | FMDOPSYS DS X | OPERATING SYSTEM |
| 000001 | FMDDSORG DS X | DSORG |
| 000002 | FMDUSAGE DS X | ALLOWABLE USAGE FOR THIS DSORG |
| 000003 | FMDRECFM DS X | DEFAULT RECORD FORMAT |
| 000004 | FMDUNIT DS X | DEFAULT UNIT |
| 000005 | DS X | RESERVED |
| 000006 | FMDBLELN DS Y | REQUIRED LEN OF BLE |
| 000008 | FMDBMVTI DS Y | OFFSET TO BMVT ENTRY FOR INPUT |
| 00000A | FMDBMVTO DS Y | OFFSET TO BMVT ENTRY FOR OUTPUT |
| 00000C | FMDDRECL DS F | DEFAULT LRECL |
| 000010 | FMDDBLKS DS F | DEFAULT BLKSIZE |
| 000014 | FMDMRECL DS F | MAXIMUM LRECL |
| 000018 | FMDMBLKS DS F | MAXIMUM BLKSIZE |
| 00001C | FMDOCSO DS Y | OFFSET TO BMVT ENTRY FOR OP/CL |
| 00001E | DS Y | RESERVED |
| 000020 | DS F | RESERVED |
| 00024 | FMDDSLEN EQU ((*-FMDDS)/4)*4 | LENGTH OF FMD TABLE ENTRY |

1.55 #FMIOWRK

```

COPY #FMIOWRK
*****
*      ---FILE MAPPING I/O MODULE BASIC WORK AREA MAP---
*
* ALL FILE MAPPING OPERATING SYSTEM DEPENDENT I/O MODULES USE THIS
* WORKAREA MAP TO ENSURE THAT ALL WORK AREAS BEGIN WITH COMMON
* STORAGE. THE INDIVIDUAL MODULES MAY EXTEND THIS DSECT AS NEEDED.
*
* THE COMMON WORK AREA MAP ALLOWS EACH MODULE TO OPERATE
* INDEPENDENTLY REGARDLESS OF THE WAY IN WHICH CONTROL MAY HAVE BEEN
* PASSED TO IT.
*
*
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|---------------------------------|
| 000000 | WRK DSECT | 08:27:14 06/26/86 |
| 000000 | WRKID DS CL8 | EYECATCHER |
| 000008 | WRKOSSAV DS 18F | OS-STYLE SAVE AREA |
| 000050 | WRKVSSAV DS 18F | OS-STYLE SAVEAREA FOR VSAM ONLY |
| 000098 | WRKSVR13 DS F | REGISTER 13 SAVE AREA |
| | WRKINP #FLAG X'80' | CALLED FOR INPUT SERVICE ONLY |
| 00009C | WRKINPI DS 0XL1 | |
| | 00080 WRKINPM EQU X'80' | |
| | WRKOUT #FLAG X'40' | CALLED FOR OUTPUT SERVICE ONLY |
| 00009C | WRKOUTI DS 0XL1 | |
| | 00040 WRKOUTM EQU X'40' | |
| | WRKIOI DS 0XL1 | CALLED FOR UPDATE SERVICE |
| 00009C | WRKIOI DS 0XL1 | |
| | 000C0 WRK1OM EQU WRKINPM+WRKOUTM | |
| 00009C | WRKFLAG1 DS X | FLAG BYTE 1 |
| 00009D | WRKFLAG2 DS X | FLAG BYTE 2 |
| 00009E | WRKFLAG3 DS X | FLAG BYTE 3 |
| 00009F | WRKFLAG4 DS X | FLAG BYTE 4 |
| 0000A0 | WRKDWD DS D | WORK DOUBLEWORD |
| 0000A8 | WRKFWD DS F | WORK FULLWORD |
| 0000AC | WRKHWD DS H | WORK HALFWORD |
| 0000AE | WRKBYTE1 DS X | WORK BYTE |
| 0000AF | WRKBYTE2 DS X | WORK BYTE |
| 0000B0 | | RESERVED |
| | DS 3F | |
| 0002F | WRKCOMLN EQU (*-WRK+3)/4 | LEN OF COMMON AREA IN FWDS |

1.56 #FSRTABL

```
COPY #FSRTABL
*****
*          *
*  DSECT FOR FUNCTION SORT TABLE HEADER (FSRTABLE)      *
*          *
*****  


| <u>Offset</u> | <u>Value</u>                                      |                                        |
|---------------|---------------------------------------------------|----------------------------------------|
| 000000        | FSRTABLE DSECT                                    | 12:06:46 04/04/85                      |
| 000000        | FSRTENT DS H                                      | NUMBER OF USED ENTRIES                 |
| 000002        | FSRTENT DS H                                      | NUMBER OF AVAILABLE ENTRIES            |
| 000004        | DS F                                              | RESERVED                               |
| 000008        | FSRTLEN EQU *-FSRTABLE                            | LENGTH OF RESPONSE TABLE HEADER        |
|               |                                                   | *****                                  |
|               | *                                                 | *                                      |
|               | *  DSECT FOR FUNCTION SORT TABLE ENTRY (FSRTNTRY) | *                                      |
|               | *                                                 | *                                      |
|               | *****                                             |                                        |
| 000000        | FSRTNTRY DSECT                                    |                                        |
| 000000        | FSRTYPE DS C                                      | FUNC TYPE (MENU,DIALOG,PROGRAM,SYSTEM) |
| 000001        | DS C                                              |                                        |
| 000002        | FSRTID DS CL8                                     | FUNCTION NAME                          |
| 00000A        | FSRTNAME DS CL8                                   | DIALOG/PROGRAM INVOKED BY FUNCTION     |
| 000012        | FSRTindx DS H                                     | FUNCTION INDEX                         |
| 00014         | FSRTELEN EQU *-FSRTNTRY                           | LENGTH OF FUNCTION SORT TABLE ENTRY    |


```

1.57 #GABDS

| COPY #GABDS | | | | | |
|--|-------------|---|--|--|----------|
| ***** | | | | | |
| *** GAB: ADS APPLICATION GENERATOR APPLICATION BLOCK *** | | | | | |
| *** | | | | | |
| *** GAB IS A DSECT THAT CONTAINS APPLICATION WIDE DATA ABOUT *** | | | | | |
| *** AN ADS APPLICATION. *** | | | | | |
| *** | | | | | |
| ***** | | | | | |
| Offset | Value | | | | |
| 000000 | GAB | DSECT | | 12/05/89 15:15:49 | 09/04/90 |
| 000000 | GABNAME | DS CL8 | | APPLICATION NAME | |
| 000008 | GABDATE | DS CL8 | | APPLICATION GEN DATE | |
| 000010 | GABTIME | DS CL8 | | APPLICATION GEN TIME | |
| 000018 | GABDESC | DS CL32 | | APPLICATION DESCRIPTION | |
| 000038 | GABTOPF | DS CL8 | | TOP FUNCTION NAME | |
| 000040 | GABSFCN | DS CL8 | | SIGNON FUNCTION NAME | |
| 000048 | GABPRDES | DS CL8 | | DEFAULT PRINT DESTINATION | |
| 000050 | GABFGTCA | DS A | | FIRST GTC POINTER | |
| 000054 | GABLGTCA | DS A | | LAST GTC POINTER | |
| 000058 | GABFGARA | DS A | | FIRST GAR POINTER | |
| 00005C | GABLGARA | DS A | | LAST GAR POINTER | |
| 000060 | GABFGRKA | DS A | | FIRST GRK POINTER | |
| 000064 | GABLGRKA | DS A | | LAST GRK POINTER | |
| 000068 | GABFGFEA | DS A | | FIRST GFE POINTER | |
| 00006C | GABLGFEA | DS A | | LAST GFE POINTER | |
| 000070 | GABFAGLA | DS A | | FIRST AGL POINTER | |
| 000074 | GABLAGLA | DS A | | LAST AGL POINTER | |
| 000078 | GABFAGBA | DS A | | FIRST AGB POINTER | |
| 00007C | GABLAGBA | DS A | | LAST AGB POINTER | |
| 000080 | GABFARDA | DS A | | FIRST ARD POINTER | |
| 000084 | GABLARDA | DS A | | LAST ARD POINTER | |
| 000088 | GABAVER | DS H | | APPLICATION VERSION NUMBER | |
| 00008A | GABTOPFX | DS H | | SIGNON FUNCTION INDEX | |
| 00008C | GABSECL | DS X | | SECURITY CLASS | |
| 00008D | GABPRCL | DS X | | DEFAULT PRINTER CLASS | |
| 00008E | GABDFMT | DS X | | DEFAULT DATE FORMAT CODE | |
| | PRINT NOGEN | | | | |
| | GABSORQ | #FLAG X'80' | | ON = SIGNON REQUIRED | |
| | GABSOOP | #FLAG X'40' | | ON = SIGNON IS OPTIONAL | |
| | GABMALL | #FLAG X'20' | | ON = MENUS CONTAIN ALL RESPONSES | |
| | GABMSEC | #FLAG X'10' | | ON = MENUS ARE SECURITY TAILORED | |
| | GABFAST | #FLAG X'08' | | This flag will NOT be needed after 10.21 | |
| | * | | | Once 10.21 source becomes final, this | |
| | * | | | flag may be reused. *MCM88081* | |
| 00008F | GABFLAG | DS X | | GAB FLAG BYTE | |
| | GABROAU | #FLAG X'40' | | ON = RESTART OPTION AUTOMATIC | |
| | GABROPR | #FLAG X'20' | | ON = RESTART OPTION PROMPT | |
| | GABROMA | #FLAG X'10' | | ON = RESTART OPTION MANUAL | |
| | * | | | ALL OFF RESTART = NO | |
| | GABONLN | #FLAG X'08' | | ON = APPLICATION WILL RUN ONLINE | |
| | GABBTC | #FLAG X'04' | | ON = APPLICATION WILL RUN BATCH | |
| | PRINT GEN | | | | |
| | *NOTE: | FOR RELEASE 10.1 ENVIRONMENT=ANY HAS BEEN DISABLED.*RQE85284* | | | |
| | * | HOWEVER, IN THE FUTURE, IF GABONLN AND GABBTC *RQE85284* | | | |
| | * | ARE BOTH ON, THEN THE APPLICATION WILL *RQE85284* | | | |
| | * | RUN IN ANY MODE (BATCH OR ONLINE), IF NEITHER IS SET THE | | | |
| | * | APPLICATION MODE WILL DEFAULT TO ONLINE. | | | |
| 000090 | GABFLAG2 | DS X | | GAB FLAG BYTE 2 | |
| 000091 | | DS X | | RESERVED | |

1.57 #GABDS

| | | | | |
|--------|----------|-----|---------------|------------------------|
| 000092 | GABMAXRS | DS | H | MAXIMUM RESPNEES |
| 000094 | | DS | 3F | RESERVED |
| 000A0 | GABLEN | EQU | *-GAB | LENGTH OF GAB |
| 00028 | GABLENF | EQU | ((*-GAB+3)/4) | LENGTH OF GAB IN WORDS |

1.58 #GARDS

```

COPY #GARDS
*****
***      GAR:  ADS APPLICATION GENERATOR APPLICATION RESPONSE   ***
*** -----
***  GAR IS A DSECT THAT DESCRIBES AN APPLICATION RESPONSE IN   ***
***  THE ADS APPLICATION GENERATOR.  DURING GENERATION, THERE    ***
***  IS A GAR BUILT FOR RESPONSE DEFINED BY THE USER.  THE RUN-  ***
***  TIME EQUIVALENT OF A GAR IS AN APRE (APPLICATION RESPONSE   ***
***  ELEMENT).  A GAR WHICH HAS A NON-ZERO GARAID FIELD ALSO    ***
***  HAS A RUNTIME CONTROL BLOCK CALLED AN RKE.                  ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------------|-----------------------------------|
| 000000 | GAR DSECT | 10:17:20 10/24/85 |
| 000000 | GARNXTA DS A | ADDR OF NEXT GAR |
| 000004 | GARPREVA DS A | ADDR OF PREVIOUS GAR |
| 000008 | GARKNXTA DS A | ADDR OF NEXT GRK (GAR WITH PFKEY) |
| 00000C | GARKPRVA DS A | ADDR OF PREVIOUS GRK |
| 000010 | GARNAME DS CL8 | RESPONSE NAME |
| 000018 | GARDDESC DS CL28 | DESCRIPTION |
| 000034 | GARDSCOF DS A | LIT POOL OFFSET TO DESCRIPTION |
| 000038 | GARIFNAM DS CL8 | NAME OF FUNCTION INVOKED |
| 000040 | GARIGFEA DS A | ADDR OF GFE FOR INVOKED FUNC |
| 000044 | GARIFKEY DS F | DBKEY OF INVOKED FUNCTION RECORD |
| 000048 | GARSFNC DS H | TRUE SYSTEM FUNCTION CODE |
| 00004A | GARINDX DS H | INDEX OF THIS GAR RELATIVE TO 0 |
| 00004C | GARKINDX DS H | INDEX OF THIS RKE RELATIVE TO 0 |
| 00004E | GARIFIX DS H | INDEX OF FUNCTION INVOKED |
| 000050 | GARAID DS X | ATTENTION IDENTIFIER (AID) BYTE |
| 000051 | GARSECL DS X | SECURITY CLASS |
| 000052 | GARDSCLN DS X | LENGTH OF DESCRIPTION |
| 000053 | GARCTRL DS X | CONTROL COMMAND |
| | GARLBL #FLAG X'80' | ON IF GLOBAL RESPONSE |
| 000054 | GARLBLI DS OXL1 | |
| | 000080 GARLBLM EQU X'80' | |
| | GARSYSF #FLAG X'40' | ON IF TRUE SYS FUNC INVOKED |
| 000054 | | |
| | 000040 GARSYSFM EQU X'40' | |
| | GARPSSF #FLAG X'20' | ON IF PSEUDO SYS FUNC INVOKED |
| 000054 | | |
| | 000020 GARPSSFM EQU X'20' | |
| | GARINTL #FLAG X'10' | ON IF INTERNAL FUNCTION INVOKED |
| 000054 | | |
| | 000010 GARINTLM EQU X'10' | |
| | GARUNDF #FLAG X'08' | ON IF UNDEFINED DLFT RESP *MDR* |
| 000054 | | |
| | 000008 GARUNDFI DS OXL1 | |
| 000054 | | |
| | 000008 GARUNDFM EQU X'08' | |
| | GARFLAG DS X | |
| | GARCLR #FLAG X'80' | ON = RETURN CLEAR *RQE85267* |
| 000055 | | |
| | 000080 GARCLRI DS OXL1 | |
| 000055 | | |
| | 000080 GARCLRM EQU X'80' | |
| | GARCONT #FLAG X'40' | ON = RETURN CONTINUE *RQE85267* |
| 000055 | | |
| | 000040 GARCONTI DS OXL1 | |
| | GARNOSV #FLAG X'20' | ON = NOSAVE *RQE85267* |
| 000055 | | |
| | 000020 GARNOSVI DS OXL1 | |
| | GARNOSVM EQU X'20' | |
| | GARNOFI #FLAG X'10' | ON = NOFINISH *RQE85267* |
| 000055 | | |
| | 000010 GARNOFII DS OXL1 | |
| | GARNOFIM EQU X'10' | |

```
          GARIMM  #FLAG X'08'          ON = EXEC RESPONSE IMMED.*RQE85267*
000055      GARIMMI DS 0XL1
00008   GARIMMM EQU X'08'
          GARDEFR #FLAG X'04'          ON = EXEC RESPONSE DEFER *RQE85288*
000055      GARDEFRI DS 0XL1
00004   GARDEFRM EQU X'04'
000055      GARFLG2 DS X           *RQE85267*
000056      DS 2X               UNUSED
000058      GARLEN EQU *-GAR      LENGTH OF GAR
00016   GARLENF EQU ((*-GAR+3)/4) LENGTH OF GAR IN WORDS
```

1.59 #GCMDS

```

COPY #GCMDS
*****
***          GCM: GENERATOR CME ELEMENT
***          GCM IS A DSECT THAT DESCRIBES THE HEADER PORTION OF AN
***          ADS/ONLINE GENERATOR CME. IMMEDIATELY FOLLOWING THE HDR
***          IS THE ACTUAL CME ITSELF TO BE INSERTED INTO THE FDB FOR
***          THE DIALOG CURRENTLY BEING GENERATED.
***          ****
*****
```

Offset Value

| | | | |
|--------|----------|-------------------|------------------------------------|
| 000000 | GCM | DSECT | 17:40:38 02/18/83 |
| 000000 | GCMNXTA | DS A | ADDR OF NEXT GCM FOR THIS RSE |
| 000004 | GCMPREVA | DS A | ADDR OF PREVIOUS GCM FOR RSE |
| 000008 | GCMSTNUM | DS F | 1ST LINE NUMBER FOR COMMAND |
| 00000C | GCMSIZE | DS H | SIZE OF THE FOLLOWING CME |
| 00000E | GCMINDEX | DS H | INCLUDED PROCESS CODE INDEX IDENT. |
| 000010 | GCMCME | DS 0F | START OF ACTUAL CME |
| 00004 | GCMLENF | EQU ((*-GCM+3)/4) | LENGTH IN WORDS OF GCM |
| 00010 | GCMLEN | EQU GCMLENF*4 | LENGTH IN BYTES OF GCM |

1.60 #GFDDS

```

COPY #GFDDS
*****
*** GFD IS A DSECT THAT DEFINES THE FIXED DIALOG BLOCK USED BY ***
*** THE ADS/ONLINE GENERATOR. THIS BLOCK CONTAINS ALL THE ***
*** INFORMATION NEEDED TO BUILD THE FDB FOR THE ADS RUNTIME ***
*** SYSTEM. ***
*****
GBLC &RELEASES ***** ADS RELEASE LEVEL *****
GBLC &INTRLSE ***** ADS RELEASE LEVEL ***** *MCM88060

```

Offset Value

| | | | |
|--------|-----------|-------------|---|
| | | ** | Set globals for Release 10.2.1 *MCM88060 |
| | | &RELEASES | SETC '1400' * SCREENS SHOULD READ 14.1 *GWG96022 |
| | | &INTRLSE | SETC '0140' * FDBS SHOULD BE MARKED W/ 0140 *GWG96022 |
| 000000 | GFD | DSECT | 04/19/90 07:50:43 04/30/91 |
| 000000 | GFDID | DS CL4 | "GFD*" |
| 000004 | GFDNAME | DS CL8 | DIALOG NAME |
| 00000C | GFDDATE | DS CL8 | DIALOG GEN DATE |
| 000014 | GFDTIME | DS CL8 | DIALOG GEN TIME |
| 00001C | GFDMPNM | DS CL8 | ONLINE MAP NAME * |
| 000024 | GFDMPDT | DS CL8 | DATE MAP COMPILED |
| 00002C | GFDMPMT | DS CL8 | TIME MAP COMPILED |
| 000034 | GFDSCHNM | DS CL8 | SCHEMA NAME |
| 00003C | GFDSSNM | DS CL8 | SUBSCHEMA NAME |
| 000044 | GFDAREC | DS CL32 | AUTOSTATUS RECORD NAME |
| 000064 | | DS F | RESERVED |
| 000068 | GFDSSDBK | DS F | DBKEY OF SUBSCHEMA RECORD |
| 00006C | GFDMPDBK | DS F | DBKEY OF MAP RECORD |
| 000070 | GFDNSSAN | DS H | # OF SUBSCHEMA AREA NAMES |
| 000072 | GFDVER | DS H | SCHEMA VERSION |
| 000074 | GFDMPVER | DS H | ONLINE MAP VERSION * |
| 000076 | GFDVER | DS H | DIALOG VERSION |
| 000078 | GFDAVER | DS H | AUTOSTATUS RECORD VERSION |
| 00007A | GFDNRECS | DS H | # OF ONLINE MAP RECORDS |
| 00007C | GFDNFLDS | DS H | # OF ONLINE MAP FIELDS |
| 00007E | GFDNDREC | DS H | # OF DIALOG RECORDS |
| 000080 | GFDRESPMI | DS H | MAP INDEX OF RESPONSE FIELD |
| 000082 | GFDRESPRI | DS H | GRD INDEX OF RESPONSE FIELD |
| 000084 | GFDRESPOF | DS H | RSP OFFSET WITHIN RECORD |
| 000086 | GFDRESPSZ | DS H | # OF BYTES IN RESP FLD |
| 000088 | GFDMSGMI | DS H | MAP INDEX OF MESSAGE FIELD |
| 00008A | GFDMSGRI | DS H | GRD INDEX OF MESSAGE FIELD |
| 00008C | GFDMSGOF | DS H | MSG OFFSET WITHIN RECORD |
| 00008E | GFDMSGSZ | DS H | # OF BYTES IN MSG FIELD |
| 000090 | GFDSEGVW | DS X | MRB - SUBSCHEMA SEGMENTED VIEW |
| | GFDMAIN | #FLAG X'80' | ON IF MAINLINE DIALOG |
| 000091 | GFDMAINI | DS 0XL1 | |
| 000080 | GFDMAINM | EQU X'80' | |
| | GFDEXCD | #FLAG X'40' | ON IF EXECUTABLE CD PROCESSES |
| 000091 | GFDEXCDI | DS 0XL1 | |
| 000040 | GFDEXCDM | EQU X'40' | |
| | GFD3279 | #FLAG X'20' | ON IF 3279 EXTENDED SCREEN |
| 000091 | GFD3279I | DS 0XL1 | |
| 000020 | GFD3279M | EQU X'20' | |
| | GFDACT | #FLAG X'10' | ON IF ACTIVITY LOGGING ENABLED |
| 000091 | GFDACI | DS 0XL1 | |
| 000010 | GFDACM | EQU X'10' | |
| | GFDCOMP | #FLAG X'08' | ON IF DIALOG INFO IS CLEAN |
| 000091 | GFDCOMPI | DS 0XL1 | |
| 00008 | GFDCOMPM | EQU X'08' | |

| | | |
|--------|--------------------------|--|
| 000091 | GFDAUTO #FLAG X'04' | ON IF AUTOSTATUS DIALOG |
| | GFDAUTOI DS OXL1 | |
| 000091 | 00004 GFDAUTOM EQU X'04' | |
| | GFDRDY #FLAG X'02' | ON FOR OLD SS, NEED AREA READIES |
| 000091 | 00002 GFDRDYM EQU X'02' | |
| | GFDNCMP #FLAG X'01' | ON IF NEW COPY/WORK INFO CLEANES |
| 000091 | 00001 GFDNCMPI DS OXL1 | |
| 000091 | GFDNCMPM EQU X'01' | |
| 000091 | GDFFLAG DS X | GFD FLAG BYTE |
| | GFDSYMT #FLAG X'80' | ON IF SYMBOL TABLE IS IN USE |
| 000092 | 00080 GFDSYMTM EQU X'80' | |
| | GFDCOBM #FLAG X'40' | ON IF USING COBOL MOVES |
| 000092 | 00040 GFDCOBMM EQU X'40' | |
| | GFDNBKP #FLAG X'20' | ON IF BACKPAGE OPTION IN USE |
| 000092 | 00020 GFDNBKPI DS OXL1 | |
| | GFDNBKPM EQU X'20' | |
| | GFDWRWS #FLAG X'10' | ON IF BROWSE (VS. UPDATE) MODE |
| 000092 | 00010 GFDWRWSI DS OXL1 | |
| | GFDWRWSM EQU X'10' | |
| | GFDENTP #FLAG X'08' | ON IF MAP ENTRY POINT*JMA85065* |
| 000092 | 00008 GFDENTPI DS OXL1 | |
| | GFDPAGM EQU X'08' | |
| 000092 | 00004 GFDPAGMI DS OXL1 | |
| | GFDPAGMM EQU X'04' | |
| | GFD\$PAG #FLAG X'02' | ON IF MAP HAS \$ PAGE FIELD |
| 000092 | 00002 GFDS\$PAGI DS OXL1 | |
| | GFDS\$PAGM EQU X'02' | |
| | GFDNTBL #FLAG X'01' | NO LNT,OFT,MDT TBLS GE*LCB84199* |
| 000092 | 00001 GFDNTBLI DS OXL1 | |
| | GFDNTBLM EQU X'01' | |
| 000092 | 00001 GDFFLAG2 DS X | GFD 2ND FLAG BYTE |
| 000093 | GFDMAPPG DS X | MAP PAGING BYTE |
| 000094 | GFDPREFIX DS CL2 | MESSAGE PREFIX *LMA86162* |
| | GFDNRLK #FLAG X'80' | ON IF NO RETRIEVAL LCK*LMA86162* |
| 000096 | 00080 GFDNRLKI DS OXL1 | |
| | GFDNRLKM EQU X'80' | |
| | GFDNADS #FLAG X'40' | ON IF NO AUTO DISPLAY DKJ86210 |
| 000096 | 00040 GFDNADSI DS OXL1 | |
| | GFDNADSM EQU X'40' | |
| 000096 | 00040 GDFFLAG3 DS X | GFD 3D FLAG BYTE *LMA86162* |
| 000097 | DS CL3 | RESERVED *LMA86162* |
| 00009A | GFDSFDDN DS CL8 | SUSPENSE FILE DDNAME |
| 0000A4 | DS OF | *JMA85071* |
| | * | BATCH INPUT MAP FIELDS FOLLOW *JMA85071* |
| 0000A4 | GFDIMNM DS CL8 | BATCH INPUT MAP NAME *JMA85071* |
| 0000AC | GFDIMDT DS CL8 | INP MAP DATE COMPILED*JMA85071* |
| 0000B4 | GFDIMTM DS CL8 | INP MAP TIME COMPILED*JMA85071* |
| 0000BC | GFDIMDBK DS F | DBKEY OF INPUT MAP RC*JMA85071* |
| 0000C0 | GFDIMVER DS H | BATCH INPUT MAP VER *JMA85071* |
| 0000C2 | GFDIMNRC DS H | # OF INP MAP RECORDS *JMA85071* |
| 0000C4 | GFDIMFLD DS H | # OF INP MAP FIELDS *JMA85071* |
| 0000C6 | GFDIMRMI DS H | INPUT MAP RESPONSE *JMA85071* |
| | * | FIELD MAP INDEX *JMA85071* |
| 0000C8 | GFDIMRRI DS H | INPUT MAP RESPONSE *JMA85071* |
| | * | FIELD GRD INDEX *JMA85071* |
| 0000CA | GFDIMROF DS H | INPUT MAP RESPONSE *JMA85071* |
| | * | OFFSET IN RECORD *JMA85071* |
| 0000CC | GFDIMRSZ DS H | # BYTES IN RESP FLD *JMA85071* |
| 0000CE | GFDIMMMI DS H | INP MAP MSG MAP INDX *JMA85071* |
| 0000D0 | GFDIMMRI DS H | INP MAP MSG GRD INDX *JMA85071* |
| 0000D2 | GFDIMMOF DS H | INP MAP MSG REC OFFST*JMA85071* |
| 0000D4 | GFDIMMSZ DS H | # BYTES IN MSG FLD *JMA85071* |

| | | | | |
|--------|----------------------------------|-----|----------|----------------------------------|
| 0000D6 | DS | CL6 | RESERVED | * |
| 0000DC | GFDIMDDN | DS | CL8 | DDNAME |
| 0000E4 | | DS | OF | *JMA85071* |
| | * BATCH OUTPUT MAP FIELDS FOLLOW | | | *JMA85071* |
| 0000E4 | GFDOMMM | DS | CL8 | BATCH OUTPUT MAP NAME*JMA85071* |
| 0000EC | GFDOMDT | DS | CL8 | OUT MAP DATE COMPILED*JMA85071* |
| 0000F4 | GFDOMTM | DS | CL8 | OUT MAP TIME COMPILED*JMA85071* |
| 0000FC | GFDOMDBK | DS | F | DBKEY OF OUTPT MAP RC*JMA85071* |
| 000100 | GFDOMVER | DS | H | BATCH OUTPUT MAP VER *JMA85071* |
| 000102 | GFDOMNRC | DS | H | # OF OUTP MAP RECORDS*JMA85071* |
| 000104 | GFDOMFLD | DS | H | # OF OUTP MAP FIELDS *JMA85071* |
| 000106 | GFDOMRMI | DS | H | OUTPUT MAP RESPONSE *JMA85071* |
| | * FIELD MAP INDEX | | | *JMA85071* |
| 000108 | GFDOMRRI | DS | H | OUTPUT MAP RESPONSE *JMA85071* |
| | * FIELD GRD INDEX | | | *JMA85071* |
| 00010A | GFDOMROF | DS | H | OUTPUT MAP RESPONSE *JMA85071* |
| | * OFFSET IN RECORD | | | *JMA85071* |
| 00010C | GFDOMRSZ | DS | H | # BYTES IN RESP FLD *JMA85071* |
| 00010E | GFDOMMMI | DS | H | OUT MAP MSG MAP INDX *JMA85071* |
| 000110 | GFDOMMRI | DS | H | OUT MAP MSG GRD INDX *JMA85071* |
| 000112 | GFDOMMDF | DS | H | OUT MAP MSG REC OFFST*JMA85071* |
| 000114 | GFDOMMSZ | DS | H | # BYTES IN MSG FLD *JMA85071* |
| 000118 | | DS | F | RESERVED *MET88074* |
| 00011C | | DS | H | RESERVED *MET88074* |
| 00011E | GFDOMDDN | DS | CL8 | DDNAME |
| 000126 | GFDSQLAM | DS | CL8 | SQL ACCESS MODULE NAME MET90342 |
| 00012E | GFDSQLSS | DS | CL8 | SQL SS NAME FOR TABLES MET90342 |
| | GFDSQ TU #FLAG X'80' | | | USA SQL TIME STANDARD *JMA91119* |
| 000136 | GFDSQLTUI | DS | 0XL1 | |
| 00080 | GFDSQ TUM | EQU | X'80' | |
| | GFDSQ TI #FLAG X'40' | | | ISO SQL TIME STANDARD *JMA91119* |
| 000136 | GFDSQLTII | DS | 0XL1 | |
| 00040 | GFDSQ TIM | EQU | X'40' | |
| | GFDSQ TE #FLAG X'20' | | | EUR SQL TIME STANDARD *JMA91119* |
| 000136 | GFDSQLTEI | DS | 0XL1 | |
| 00020 | GFDSQ TEM | EQU | X'20' | |
| | GFDSQ TJ #FLAG X'10' | | | JIS SQL TIME STANDARD *JMA91119* |
| 000136 | GFDSQLTJI | DS | 0XL1 | |
| 00010 | GFDSQ TJM | EQU | X'10' | |
| 000136 | GFDSQLTM | DS | X | SQL TIME FORMAT MET90342 |
| | GFDSQ DU #FLAG X'80' | | | USA SQL DATE STANDARD *JMA91119* |
| 000137 | GFDSQLDUI | DS | 0XL1 | |
| 00080 | GFDSQ DUM | EQU | X'80' | |
| | GFDSQ DI #FLAG X'40' | | | ISO SQL DATE STANDARD *JMA91119* |
| 000137 | GFDSQLDII | DS | 0XL1 | |
| 00040 | GFDSQ DIM | EQU | X'40' | |
| | GFDSQ DE #FLAG X'20' | | | EUR SQL DATE STANDARD *JMA91119* |
| 000137 | GFDSQLDEI | DS | 0XL1 | |
| 00020 | GFDSQ DEM | EQU | X'20' | |
| | GFDSQ DJ #FLAG X'10' | | | JIS SQL DATE STANDARD *JMA91119* |
| 000137 | GFDSQLDJI | DS | 0XL1 | |
| 00010 | GFDSQ DJM | EQU | X'10' | |
| 000137 | GFDSQLDT | DS | X | SQL DATE FORMAT MET90342 |
| | GFDANSI #FLAG X'80' | | | ANSI STD SQL PARSED *JMA91119* |
| 000138 | GFDANSII | DS | 0XL1 | |
| 00080 | GFDANSIM | EQU | X'80' | |
| | GFDIPS #FLAG X'40' | | | FIPS STD SQL PARSED *JMA91119* |
| 000138 | GFDIPSI | DS | 0XL1 | |
| 00040 | GFDIPSM | EQU | X'40' | |
| | GFDDB2 #FLAG X'20' | | | DB2 STD SQL PARSED *JMA91119* |
| 000138 | GFDDB2I | DS | 0XL1 | |
| 00020 | GFDDB2M | EQU | X'20' | |
| | GFDSQL2 #FLAG X'10' | | | SQL2 STD SQL PARSED *JMA91119* |
| 000138 | GFDSQL2I | DS | 0XL1 | |
| 00010 | GFDSQL2M | EQU | X'10' | |

| | | | | | | |
|--------|----------|-------------------|---------------------|-----|------------|-----------------------------|
| | GFDSAAM | #FLAG X'08' | SAA | STD | SQL PARSED | *JMA91119* |
| 000138 | GFDSAAI | DS 0XL1 | | | | |
| 00008 | GFDSQLFL | EQU X'08' | | | | |
| 000138 | GFDSAAI | DS X | SQL COMPLIANCE FLAG | | | MET90342 |
| 000139 | | DS X | RESERVED | | | MET90342 |
| 00013A | | DS CL20 | RESERVED | | | MET90342 |
| 00014E | | DS H | RESERVED | | | *MET88074* |
| 000150 | | DS 4F | RESERVED | | | *MET88074* |
| 00160 | GFDLLEN | EQU *-GFD | | | | LENGTH OF GFD FIXED PORTION |
| 00058 | GFDLLENF | EQU ((*-GFD+3)/4) | | | | LENGTH OF GFD IN WORDS |

1.61 #GFEDS

```

COPY #GFEDS
*****
***      GFE:  ADS APPLICATION GENERATOR FUNCTION ELEMENT
***      GFE IS A DSECT THAT DESCRIBES AN APPLICATION FUNCTION IN
***      THE ADS APPLICATION GENERATOR.  DURING GENERATION, THERE
***      IS A GFE BUILT FOR EACH MENU, DIALOG OR MENU/DIALOG FUNCTION
***      THE RUNTIME EQUIVALENT OF A GFE IS AN FDE (FUNCTION DEFINI-
***      TION ELEMENT).
***      ****
*****
```

Offset Value

| | | | | |
|--------|----------|-------------------|--------------------------------|------------|
| 000000 | GFE | DSECT | 12:16:03 04/04/85 | 04/20/95 |
| 000000 | GFENXTA | DS A | ADDR OF NEXT GFE | |
| 000004 | GFPREVA | DS A | ADDR OF PREVIOUS GFE | |
| 000008 | GFEFGFRA | DS A | ADDR OF FIRST GFR FOR FUNCTION | |
| 00000C | GFEFGFRA | DS A | ADDR OF LAST GFR FOR FUNCTION | |
| 000010 | GFEFGMEA | DS A | ADDR OF FIRST GME FOR FUNCTION | |
| 000014 | GFEFGMEA | DS A | ADDR OF LAST GME FOR FUNCTION | |
| 000018 | GFEFMSQA | DS A | ADDR OF FIRST MSQ FOR FUNCTION | |
| 00001C | GFEFMSQA | DS A | ADDR OF LAST MSQ FOR FUNCTION | |
| 000020 | GFEFURNA | DS A | ADDR OF FIRST URN FOR FUNCTION | |
| 000024 | GFEFURNA | DS A | ADDR OF LAST URN FOR FUNCTION | |
| 000028 | GFEKGARA | DS A | ADDR OF DEFAULT RESPONSE GAR | |
| 00002C | GFESEQN | DS F | GFE SEQUENCE NUMBER | |
| 000030 | GFEDESC | DS CL28 | FUNCTION DESCRIPTION | ADS370/335 |
| 00004C | | DS CL4 | AVAILABLE | ADS370/335 |
| 000050 | GFEFDE | DS 0F | START OF FDE | |
| 00050 | GFELEN | EQU *~-GFE | LENGTH OF GFE | |
| 00014 | GFELENF | EQU ((*-GFE+3)/4) | LENGTH OF GFE IN WORDS | |

1.62 #GFRDS

```
COPY #GFRDS
*****
***      GFR:  ADS APPL GENERATOR FUNCTION/RESPONSE JUNCTION      ***
***  -----
***  GFR IS A DSECT THAT IS THE INTERNAL CONTROL BLOCK USED TO      ***
***  CONNECT A RESPONSE VALID FOR A FUNCTION TO THE GAR FOR          ***
***  THE RESPONSE.                                                 ***
*****
```

Offset Value

| | | | |
|--------|----------|-------------------|-----------------------------|
| 000000 | GFR | DSECT | 12:57:53 04/04/85 |
| 000000 | GFRNXTA | DS A | ADDR OF NEXT GFR |
| 000004 | GFRPREVA | DS A | ADDR OF PREVIOUS GFR |
| 000008 | GFRGARA | DS A | GAR ADDR FOR RESPONSE |
| 0000C | GFRLEN | EQU *-GFR | LENGTH OF GFR |
| 00003 | GFRLENF | EQU ((*-GFR+3)/4) | LENGTH OF GFR IN FULL WORDS |

1.63 #GLEDS

```
COPY #GLEDS
*****
***          GLE:  GENERATOR LITERAL POOL ELEMENT
***          GLE IS A DSECT THAT DESCRIBES THE HEADER PORTION OF AN
***          ADS/ONLINE GENERATOR LITERAL POOL ELEMENT. FOLLOWING THE
***          HDR IS THE ACTUAL LITERAL.
***          ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------|
| 000000 | GLE | DSECT |
| 000000 | GLENXTA | DS A |
| 000004 | GLEPREVA | DS A |
| 000008 | GLESIZE | DS H |
| 00000A | GLELPOFF | DS H |
| 000003 | GLELEN | EQU ((*-GLE+3)/4) |
| 00000C | GLELIT | DS 0X |

19:34:16 12/16/80
ADDR OF NEXT GLE
ADDR OF PREVIOUS GLE
SIZE OF THE FOLLOWING LITERAL
ACTUAL OFFSET INTO LIT POOL
LENGTH IN WORDS OF GLE
START OF LITERAL

1.64 #GMDDS

```

COPY #GMDDS
*****
*** GMD: GENERATOR MODULE TABLE ELEMENT ***
*** GMD IS A DSECT THAT DESCRIBES A PROCESS INCLUDED VIA ***
*** AN 'INCLUDE' COMMAND IN ANOTHER PREMAP OR RESPONSE PROCESS. ***
*** THIS INFORMATION IS USED TO KEEP TRACK OF DEBUGGING ***
*** INFORMATION FOR NEW RUN-TIME DEBUGGING SCREEN. ***
*****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------------|---------------------------------------|
| 000000 | GMD DSECT | 13:57:40 07/27/84 05/22/91 |
| 000000 | GMDKEY DS 0CL34 | COMBINED NAME AND VERSION KEY 12.0 |
| 000000 | GMDNAME DS CL32 | NAME OF THE INCLUDED PROCESS |
| 000020 | GMDVERS DS H | VERSION OF THE INCLUDED PROCESS |
| 000022 | GMDINDEX DS H | INDEX OF THIS GCM |
| 000024 | GMDNXTA DS A | ADDRESS OF NEXT GMD |
| 000028 | GMDDBK DS F | DBKEY OF SAVED TEXT-088 RECORD |
| 00002C | GMDPRNDX DS H | INDEX OF THE PREVIOUS GCM |
| | GMDUSED #FLAG X'80' | ON WHEN PROCESS CAN'T BE INCLUDED- |
| 00002E | GMDUSEDI DS 0XL1 | |
| 000080 | GMDUSEDM EQU X'80' | (CAN'T INCLUDE PROCESS WITHIN ITSELF) |
| | * | ON WHEN EOF HIT FOR THIS PROCESS |
| 00002E | GMDEOF #FLAG X'40' | |
| 00002E | GMDEOFI DS 0XL1 | |
| 00040 | GMDEOFM EQU X'40' | |
| 00002E | GMDFLAG1 DS X | FLAG BYTE 1 |
| 00002F | DS X | UNUSED |
| 000030 | DS 0F | |
| 000030 | GMDDATCR DS CL8 | DATE CREATED |
| 000038 | GMDDATLU DS CL8 | DATE LAST UPDATED |
| 000040 | GMDNCME DS H | NUMBER OF CME'S IN THIS PROCESS |
| 000042 | GMDNINC DS H | NUMBER OF TIMES PROCESS INCLUDED |
| 000044 | GMDDMBK DS F | DBKEY OF MODULE-067 |
| 000048 | GMDSNXTA DS F | ADDRESS OF NEXT SORTED GMD 12.0 |
| 00013 | GMDLENF EQU ((*-GMD+3)/4) | LENGTH IN WORDS OF GMD |
| 0004C | GMDLEN EQU GMDLENF*4 | LENGTH IN BYTES OF GMD |

1.65 #GMEDS

```
COPY #GMEDS
*****
***      GME:  ADS APPLICATION GENERATOR MENU HEADER (INTERNAL) ***
***      GME IS A DSECT THAT IS THE INTERNAL CONTROL BLOCK USED TO ***
***      HOLD A MENU HEADING FOR A FUNCTION WHICH IS A MENU.        ***
*****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------|
| 000000 | GME | DSECT |
| 000000 | GMENXTA | DS A |
| 000004 | GMEPREVA | DS A |
| 000008 | GMECMTEO | DS H |
| 00000C | GMEMTE | DS 0F |
| 0000C | GMELLEN | EQU *-GME |
| 00003 | GMELLENF | EQU ((*-GME+3)/4) |

12:16:21 04/04/85
ADDR OF NEXT GME
ADDR OF PREVIOUS GME
MENU TBL OFFSET TO THIS MTE
START OF EMBEDDED MTE IN GME
LENGTH OF GME
LENGTH OF GME IN FULL WORDS

1.66 #GPRDS

```

COPY #GPRDS
*****
*** GPR IS A DSECT DEFINING THE ADS GENERATOR EQUIVALENT FOR THE ***
*** RUNTIME SYSTEM DCL, PME, AND RSE.                                ***
***                                                               ***
***   THE ADDITIONAL FIELDS NEEDED FOR THE GENERATOR RSE ARE      ***
***   SAVED AFTER THE COMMON PROCESS FIELDS.                            ***
***                                                               ***
***   DCL=DECLARATION  0..1 PER DIALOG.                             ***
***   PME=PREMAP        0..1 PER DIALOG.                             ***
***   RSE=RESPONSE      0..N PER DIALOG.                            ***
*****

```

| <u>Offset</u> | <u>Value</u> | | | |
|---------------|--------------|-------------------------|---|----------|
| 000000 | GPR | DSECT | 03/04/88 18:37:49 | 05/22/91 |
| 000000 | GRS | DS 0F | START OF GRS | |
| 000000 | GPRID | DS CL4 | "GDC*" OR "GPM*" OR "GRS*" | |
| 000004 | GPRNAME | DS CL32 | NAME OF THE PROCESS | |
| 000024 | GPRDATLU | DS CL8 | MODULE-067 DATE LAST UPDATED | |
| 00002C | GPRDATCR | DS CL8 | MODULE-067 DATE CREATED | |
| 000034 | GPRFGRLA | DS A | PTS TO ADDR OF FIRST PROCESS GRD | |
| 000038 | GPRLGRLA | DS A | PTS TO ADDR OF LAST PROCESS GRD | |
| 00003C | GPRFGCMA | DS A | ADDR OF FIRST GCM FOR PROCESS | |
| 000040 | GPRLGCM | DS A | ADDR OF LAST GCM FOR PROCESS | |
| 000044 | GPRFGMDA | DS A | ADDR OF FIRST GMD FOR PROCESS | |
| 000048 | GPRLGMDA | DS A | ADDR OF LAST GMD FOR PROCESS | |
| 00004C | GPRSGMDA | DS A | ADDR OF FIRST SORTED GMD | 12.0 |
| 000050 | GPRFGVCA | DS A | ADDR OF FIRST GVC FOR PROCESS | |
| 000054 | GPRLGVCA | DS A | ADDR OF LAST GVC FOR PROCESS | |
| 000058 | GPRRATA | DS A | ADDR OF READY TABLE FOR PROCESS | |
| 00005C | GPRNSTMT | DS F | TOTAL NUMBER OF STATEMENTS | |
| 000060 | GPRRUNSZ | DS F | RUNTIME LENGTH OF ALL CME'S | |
| 000064 | GPRMODKY | DS F | DBKEY OF MODULE-067 RECORD | |
| 000068 | GPRFCMEA | DS F | OFFSET TO FIRST CME | |
| 00006C | GPRPVER | DS H | PROCESS VERSION | |
| 00006E | GPRNRAT | DS H | NUMBER OF READY TABLE ENTRIES | |
| 000070 | GPRNCME | DS H | NUMBER OF CME'S IN PROCESS (CME'S FROM GMD'S NOT INCLUDED) | |
| 000072 | * | DS H | RESERVED FOR FUTURE USE | |
| | 00050 | GPRDPLEN EQU *-GPRDATLU | # BYTES TO MOVE FOR DUPLICATE RP | |
| | | GPRCOMP #FLAG X'80' | ON IF PROCESS HAS BEEN COMPILED | |
| 000074 | | GPRCOMPI DS OXL1 | | |
| | 00080 | GPRCOMPM EQU X'80' | | |
| | | GPRDEL #FLAG X'40' | ON IF PROCESS HAS BEEN DELETED | |
| 000074 | | GPRDELI DS OXL1 | | |
| | 00040 | GPRDELM EQU X'40' | | |
| | | GPREXTM #FLAG X'20' | ON IF PROCESS HAS EXTENDED MAP CMD | |
| 000074 | | GPREXTMI DS OXL1 | | |
| | 00020 | GPREXTMM EQU X'20' | | |
| | | GPRRDY #FLAG X'10' | ON IF NO AUTOREADY | |
| 000074 | | GPRRDYI DS OXL1 | | |
| | 00010 | GPRRDYM EQU X'10' | | |
| | | GPRALOW #FLAG X'08' | ON IF AUTO-EDIT ERRORS ALLOWED | |
| 000074 | | GPRALOWI DS OXL1 | | |
| | 00008 | GPRALOWM EQU X'08' | | |
| | | GPRNCMP #FLAG X'04' | ON WHEN PROCESS FAILS TO COMPILE | |
| 000074 | | GPRNCMPI DS OXL1 | | |
| | 00004 | GPRNCMPM EQU X'04' | | |
| | | GPRRDYA #FLAG X'02' | ON IF READY ALL CMD IN PROCESS | |
| 000074 | | GPRRDYAI DS OXL1 | | |
| | 00002 | GPRRDYAM EQU X'02' | | |
| | | GPRDBC #FLAG X'01' | ON IF A DB DML COMMAND WAS CODED | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|-------------------------------------|
| 000074 | GPRDBCI DS 0XL1 | |
| 00001 | GPRDBCM EQU X'01' | |
| | *PRDBC #FLAG X'01' | ON IF A DB DML COMMAND ADS370/582 |
| 000074 | GPRFLAG1 DS X | FLAG BYTE |
| 00054 | GPRDLEN EQU ((*-GPRDATLU+3)/4)*4 | # BYTES TO CLEAR WHEN DEL PREMAP |
| 00078 | GPRLEN EQU ((*-GPR+3)/4)*4 | GPR LENGTH IN BYTES |
| 0001E | GPRLENF EQU ((*-GPR+3)/4) | GPR LENGTH IN WORDS |
| | * THE FOLLOWING FIELDS ARE NEEDED ONLY FOR A RESPONSE PROCESS. | |
| 000078 | GPRNEXTA DS A | ADDR OF NEXT GRS IN DIALOG |
| 00007C | GPRPREVA DS A | ADDR OF PREVIOUS GRS |
| 000080 | GPRSREA DS A | ADDR OF CORRESPONDING SRE |
| 000084 | GPRFUNC DS CL32 | FUNCTION CODE FOR RESPONSE |
| 0000A4 | GPRFUNLN DS X | LENGTH OF FUNCTION CODE |
| 0000A5 | GPRPFKEY DS X | PFKEY FOR RESPONSE |
| | GPRRDUP #FLAG X'80' | ON IF RP DUPLICATES EXISTING RP |
| 0000A6 | GPRRDUPI DS 0XL1 | |
| 00080 | GPRRDUPM EQU X'80' | |
| | GPRDEF #FLAG X'40' | ON IF THIS IS DEFAULT RP *JMA85077* |
| 0000A6 | GPRDEFI DS 0XL1 | |
| 00040 | GPRDEFM EQU X'40' | |
| 0000A6 | GPRRFLG1 DS X | FLAG BYTE FOR RESPONSE PROCESS |
| 0000A7 | GPRRBCE1 DS X | BATCH CONTROL ELVENT FOR BATCH RSP |
| 000A8 | GRSLEN EQU ((*-GRS+3)/4)*4 | LENGTH OF GRS IN BYTES |
| 0002A | GRSLENF EQU ((*-GRS+3)/4) | LENGTH OF GRS IN WORDS |

1.67 #GRDDS

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|----------------------------------|
| 000000 | GRD DSECT | 03/24/88 20:45:58 03/11/91 |
| 000000 | GRDID DS CL4 | "GRD*" OR "GLR*" |
| 000004 | GRDNXTA DS A | ADDRESS OF NEXT GRD |
| 000008 | GRDPREVA DS A | ADDRESS OF PREVIOUS GRD |
| 00000C | GRDNAME DS CL32 | NAME OF RECORD |
| 00002C | GRDAREAN DS CL32 | NAME OF AREA RECORD IS IN |
| 00004C | GRDFGREA DS A | ADDRESS OF FIRST GRE FOR GRD |
| 000050 | GRDINTVA DS A | ADDR OF INITIALIZED RECORD |
| 000054 | GRDRECL DS F | LENGTH OF RECORD |
| 000058 | GRDLOK DS F | PRIVACY LOCKS |
| 00005C | GRDDBKEY DS F | DBKEY OF RECSYN-079 |
| 000060 | GRDDEL DS H | DELETE LOCKS |
| 000000 | GRDDEL1 EQU X'0000' | ANYTHING OK |
| 000001 | GRDDEL2 EQU X'0001' | ERASE NOT ALLOWED |
| 000002 | GRDDEL3 EQU X'0002' | STORE NOT ALLOWED |
| 000004 | GRDDEL4 EQU X'0004' | MODIFY NOT ALLOWED |
| 000062 | GRDVER DS H | RECORD VERSION NUMBER |
| 000064 | GRDMINDX DS H | ONLINE MAP RECORD INDX*JMA85074* |
| 000066 | GRDIMNDX DS H | INPUT MAP RECORD INDX*JMA85074* |
| 000068 | GRDOMNDX DS H | OUTPUT MAP RECORD INDX*JMA85074* |
| 00006A | GRDRTABO DS H | VRE OFFSET |
| | GRDMAP #FLAG X'80' | REC USED IN ONLINE MAP*JMA85074* |
| 00006C | GRDMAPI DS OXL1 | |
| 000080 | GRDMAPM EQU X'80' | RECORD USED IN PRE-MAP PROCESS |
| 00006C | GRDPRMP #FLAG X'40' | |
| 000040 | GRDPRMPI DS OXL1 | |
| | GRDPRMPM EQU X'40' | |
| | GRDRESP #FLAG X'20' | RECORD USED IN RESPONSE |
| 00006C | GRDRESP1 DS OXL1 | |
| 000020 | GRDRESPM EQU X'20' | |
| | GRDOKSR #FLAG X'10' | Record valid for Send/Receive |
| 00006C | GRDOKSRI DS OXL1 | |
| 000010 | GRDOKSRM EQU X'10' | |
| | GRDBIND #FLAG X'08' | RECORD IN SUBSCHEMA |
| 00006C | GRDBINDI DS OXL1 | |
| 000008 | GRDBINDM EQU X'08' | |
| | GRDIMAP #FLAG X'04' | REC USED IN INPUT MAP *JMA85074* |
| 00006C | GRDIMAPI DS OXL1 | |
| 000004 | GRDIMAPM EQU X'04' | |
| | GRDOMAP #FLAG X'02' | REC USED IN OUTPUT MAP*JMA85074* |
| 00006C | GRDOMAPI DS OXL1 | |
| 000002 | GRDOMAPM EQU X'02' | |
| 00006C | GRDFLG1 DS X | |
| | GRDFRSH #FLAG X'80' | REQUIRES NEW COPY IN BUFFER |
| 00006D | GRDFRSHI DS OXL1 | |
| 000080 | GRDFRSHM EQU X'80' | |
| | GRDROLE #FLAG X'40' | RECORD IS REALLY A ROLE |
| 00006D | GRDROLEI DS OXL1 | |
| 000040 | GRDROLEM EQU X'40' | |

| | | | |
|--------|--|--|---|
| | GRDPSR | #FLAG X'20' | RECORD IN LOGICAL RECORD |
| 00006D | GRDPSRI | DS 0XL1 | |
| 00020 | GRDPSRM | EQU X'20' | |
| | GRDPUP | #FLAG X'10' | RECORD PASSED FROM USER PGM |
| 00006D | GRDPUPI | DS 0XL1 | |
| 00010 | GRDPUPM | EQU X'10' | |
| | GRDASR | #FLAG X'08' | RECORD IS AUTOSTATUS RECORD |
| 00006D | GRDASRI | DS 0XL1 | |
| 00008 | GRDASRM | EQU X'08' | |
| | GRDWORK | #FLAG X'04' | RECORD IS WORK RECORD |
| 00006D | GRDWORKI | DS 0XL1 | |
| 00004 | GRDWORKM | EQU X'04' | |
| | GRDPROL | #FLAG X'02' | PRIMARY RECORD FOR A ROLE |
| 00006D | GRDPROLI | DS 0XL1 | |
| 00002 | GRDPROLM | EQU X'02' | |
| 00006D | GRDFLG2 | DS X | |
| 00006E | GRDCRECL | DS H | COMPRESSED INIT REC SIZE |
| 000070 | GRDUCNT | DS H | GENERATOR USE COUNT |
| 000072 | GRDINDEX | DS H | GENERATOR RECORD INDEX |
| | GRDPRNC | #FLAG X'80' | ON IF OLD NEW COPY RECORD |
| 000074 | GRDPRNCI | DS 0XL1 | |
| 00080 | GRDPRNCM | EQU X'80' | |
| | GRDPRWK | #FLAG X'40' | ON IF OLD WORK RECORD |
| 000074 | GRDPRWKI | DS 0XL1 | |
| 00040 | GRDPRWKM | EQU X'40' | |
| | GRDSEG | #FLAG X'20' | ON IF SS VIEW DIFFERS FROM MAP |
| 000074 | GRDSEGI | DS 0XL1 | |
| 00020 | GRDSEGM | EQU X'20' | |
| | GRDBLT | #FLAG X'10' | ON IF GRD HAS BEEN PROCESSED |
| 000074 | GRDBLTI | DS 0XL1 | |
| 00010 | GRDBLTM | EQU X'10' | |
| | GRDINT | #FLAG X'08' | INTERNAL RECORD *JMA90351* |
| 000074 | GRDINTI | DS 0XL1 | |
| 00008 | GRDINTM | EQU X'08' | |
| | GRDSQLT | #FLAG X'04' | CREATD FROM SQL TABLE*JMA91065* |
| 000074 | GRDSQLTI | DS 0XL1 | |
| 00004 | GRDSQLTM | EQU X'04' | |
| 000074 | GRDGFLAG | DS X | GENERATOR FLAG BYTE |
| 000075 | | DS X | Reserved *MCM88064 |
| | *** EXTENSION FOR LOGICAL RECORD SUPPORT IN ADS/ONLINE GENERATOR | | |
| 000076 | GRDNRECS | DS H | NUMBER OF RECORDS IN LR |
| 000078 | GRDNPSRA | DS A | NEXT GRD IN LOGICAL RECORD |
| 00007C | GRDPPSRA | DS A | PRIOR GRD IN LOGICAL RECORD |
| 000080 | GRDNGLRA | DS A | NEXT GLR IN LOGICAL REC CHAIN |
| 000084 | GRDFGVCA | DS A | ADDR OF FIRST GVC FOR LR |
| 000088 | GRDLGVCA | DS A | ADDR OF LAST GVC FOR LR |
| | *** EXTENSION FOR ROLE SUPPORT IN ADS/ONLINE GENERATOR | | |
| 00008C | ORG | GRDNGLRA | |
| 000080 | GRDPRECA | DS A | ADDR OF GRD FOR PRIMARY REC |
| 000084 | ORG | | |
| | *** Added following fields to support SEND/RECEIVE FDE | *MCM88064 | |
| | *** generation in Release 10.21. | *MCM88064 | |
| 00008C | GRDFDSIZ | DS F | Initialized to -1. |
| | * | -1 ==> This record not yet processed | |
| | * | for FDE generation. | |
| | * | ==> Number of bytes needed for | |
| | * | FDEs for this record. | |
| | * | Does NOT include the FDH. | |
| 000090 | GRDFDO | DS F | Initialized to -1. |
| | * | -1 ==> FDEs have not yet been | |
| | * | requested for this GRD. | |
| | * | ==> Offset into FDE area in FDB where | |
| | * | FDEs for this record will begin. | |
| 000094 | GRDFDNXT | DS A | Address of next GRD/GRE for which we need |
| | * | to generate FDEs. Linked in same order | |

| | | | |
|--------|-------------|-----------------|--|
| 000098 | * | | they will appear in FDB. |
| | GRDOFLDS DS | F | Number of fields in record. |
| 00009C | * | | |
| | GRDSCHML DS | H | len of schema name (in GRDSCHMN) *JMA91067* |
| 00009E | * | | (when grd creatd from SQL TABLE) *JMA91067* |
| | GRDSCHMN DS | CL18 | schema name used with the table name*JMA91067* |
| 0000B0 | * | | (when grd creatd from SQL TABLE) *JMA91067* |
| | DS | 5F | Reserved for future expansion. |
| | * | | |
| 000C4 | GRDLEN EQU | ((*-GRD+3)/4)*4 | GRD LENGTH IN BYTES |
| 00031 | GRDLENF EQU | ((*-GRD+3)/4) | GRD LENGTH IN WORDS |

1.68 #GREDS

```

COPY #GREDS
*****
***   GRE IS A DSECT THAT DEFINES A RECORD ELEMENT ENTRY IN THE   ***
***   GWA OF THE ADS GENERATOR.  GRE'S ARE CHAINED FROM THE GRD   ***
***   FOR THE RECORD.                                              ***
***   ADDRESS OF NEXT GRE
***   ELEMENT NAME
***   LR KEYWORD VERB NAME
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------------|-------------------------------|
| 000000 | GRE DSECT | 03/24/88 20:45:37 09/04/90 |
| 000000 | GREID DS CL4 | "GRE*" |
| 000004 | GRENXTA DS A | ADDRESS OF NEXT GRE |
| 000008 | GRENAME DS CL32 | ELEMENT NAME |
| 000028 | GREKWVRB DS CL8 | LR KEYWORD VERB NAME |
| 000030 | ORG GREKWVRB | |
| 000028 | GREFGCEA DS A | ADDR OF FIRST GCE, IF ANY |
| 00002C | REGMLA DS A | ADDR OF FIRST GML, IF ANY |
| 000030 | GRELGMLA DS A | ADDR OF LAST GML IN CHAIN |
| 000034 | REGMLN0 DS H | NUMBER OF GMLS IN CHAIN |
| 000036 | GREMFIND DS H | MAP FIELD INDEX |
| 000038 | GREOFSET DS H | OFFSET TO FIELD |
| 00003A | GRELGTH DS H | LENGTH OF FIELD |
| 00003C | GREBOFF DS H | BIT OFFSET FOR BIT FIELD |
| 00003E | GREBLTH DS H | BIT FIELD LENGTH |
| 000040 | GREUSE DS H | USAGE |
| 000042 | GREOCC DS H | NUMBER OF OCCURRENCES |
| 000044 | GREOCLVL DS H | OCCURS LEVEL |
| 000046 | GREDECNM DS H | NUMBER OF DECIMAL POSITIONS |
| 000048 | GREDRVTO DS H | DOPE VECTOR REC TBL OFFSET |
| 00004A | GREDVROF DS H | DOPE VECTOR RECORD OFFSET |
| 00004C | GREMILOF DS H | LIT POOL OFFSET TO MIL |
| 00004E | GREDATYP DS X | DATA TYPE FOR RDCEVAL |
| | GRECOND #FLAG X'80' | ON IF GCE'S EXIST FOR GRE |
| 00004F | GRECONDI DS 0XL1 | |
| | 00080 GRECOND M EQU X'80' | |
| | GREDPND #FLAG X'40' | ON IF FIELD IS DEPENDING ON |
| 00004F | GREDPNDI DS 0XL1 | |
| | 00040 GREDPNDM EQU X'40' | |
| | GREDVBI #FLAG X'20' | ON IF DOPE VECTOR EXISTS |
| 00004F | GREDVBI DS 0XL1 | |
| | 00020 GREDVBM EQU X'20' | |
| | GRERQST #FLAG X'10' | FOR LOGICAL RECORD PROCESSING |
| 00004F | GRERQSTI DS 0XL1 | |
| | 00010 GRERQSTM EQU X'10' | |
| | GREMFLD #FLAG X'08' | ON IF ELEMENT IS MAP FIELD |
| 00004F | GREMFLDI DS 0XL1 | |
| | 00008 GREMFLDM EQU X'08' | |
| | GREMIL #FLAG X'04' | ON IF MIL IS IN LIT POOL |
| 00004F | GREMILI DS 0XL1 | |
| | 00004 GREMILM EQU X'04' | |
| | GREXTRA #FLAG X'02' | ON IF GRE HAS GREOCGRE |
| 00004F | GREXTRAI DS 0XL1 | |
| | 00002 GREXTRAM EQU X'02' | |
| | GREKWD #FLAG X'01' | ON IF KEYWORD |
| 00004F | GREKWDI DS 0XL1 | |
| | 00001 GREKWDM EQU X'01' | |
| 00004F | GREFLAG DS X | |
| 000050 | GREDRlvl DS CL2 | RECORD LEVEL |
| 000052 | GRERDLVL DS H | Redefines Level *MCM88064 |

```

000054          DS   F             RESERVED           *LCB84132*
000058          GREOCGRE DS  0F             ADDRESS OF HIGHER GRE, IF ONE
00058  GRELEN  EQU  ((*-GRE+3)/4)*4 LENGTH OF GRE IN BYTES
00016  GRELENF EQU  ((*-GRE+3)/4) LENGTH OF GRE IN WORDS
000058          DS   F             GREOCGRE SPACE    *LCB84132*
00005C          GREPICDA DS  F             PICTURE DSECT ADDRESS *LCB84132*
000060          GRESDIG DS  X             # OF SOURCE DIGITS  *LCB84132*
000061          GRESDEC DS  X             # OF SOURCE DECIMALS *LCB84132*
000062          DS   H             RESERVED
*** Added following fields to support SEND/RECEIVE FDE      *MCM88064
*** generation in Release 10.21.                           *MCM88064
000064          GREFDSIZ DS  F             Initialized to -1.
*                               -1 ==> Record containing this GRE
*                               has never been processed
*                               for FDE generation.
*                               -1 ==> Number of bytes needed for
*                               the FDEs for this field.
*                               Does NOT include the FDH.
000068          GREFDO  DS  F             Initialized to -1.
*                               -1 ==> This GRE has never been processed
*                               for FDE generation.
*                               -1 ==> Offset into FDE area in FDB where
*                               FDEs for this field will begin.
00006C          GREFDNXT DS  A             Address of next GRD/GRE for which we need
*                               to generate FDEs. Linked in same order
*                               they will appear in FDB.
000070          GREOFLDS DS  F             Number of fields in group field.
*                               0 for non-group field.
*
GREOKSR  #FLAG X'80'   Element valid for Send/Receive
000074          GREOKSRI DS  0XL1
00080  GREOKSRM EQU  X'80'
000074          GREFLAG2 DS  X             Flag byte
000075          DS   X             RESERVED FOR FUTURE EXPANSION.
000076          GRESQDTY DS  H             SQL DATA TYPE          **SQL**
000078          GRESQPRE DS  H             SQL PRECISION        **SQL**
00007A          GRESQSCA DS  H             SQL SCALE            **SQL**
00007C          GRESQLEN DS  H             SQL LENGTH           **SQL**
00007E          DS   2X             RESERVED FOR FUTURE EXPANSION.
000080          DS   4F             Reserved for future expansion.
*
000090  GRELEN2  EQU  ((*-GRE+3)/4)*4      ADSO LEN OF GRE IN BYTES
00024  GRELENF2 EQU  ((*-GRE+3)/4)      ADSO LEN OF GRE IN WORDS
*****
*** GCE IS A DSECT WHICH DEFINES A CONDITION NAME. IT IS ***
*** CHAINED FROM THE GRE WHEN A CONDITION NAME EXISTS FOR AN ***
*** ELEMENT.                                              ***
*****
000000  GCE   DSECT
000000  GCEID DS  CL4             "GCE*"
000004  GCENXTA DS  A             ADDRESS OF NEXT GCE FOR GRE
000008  GCENAME DS  CL32            CONDITION NAME
000028  GCEGREA DS  A             ADDRESS OF GRE
00002C  GCEFGVEA DS  A             ADDR OF 1ST VALUE BLOCK (NEW GCE)
000030  GCEASR  #FLAG X'80'          ON IF 88-LVL FOR AUTOSTATUS REC
000030  GCEASRI DS  0XL1
00080  GCEASRM EQU  X'80'
000030  GCEFLAG DS  X             FLAG BYTE
000031  DS   3X             RESERVED
000034  GCELEN  EQU  ((*-GCE+3)/4)*4      LENGTH OF GCE IN BYTES
00000D  GCELENF EQU  ((*-GCE+3)/4)      LENGTH OF GCE IN WORDS
*****
*** GVE IS A DSECT WHICH DEFINES A VALUE OR RANGE OF VALUES ***
*** WHICH ARE VALID FOR A CONDITION NAME. THERE IS ONE GVE ***
*** CHAINED TO THE GCE FOR EACH VALUE/RANGE OF VALUES ALLOWED. ***

```

```
*****
000000 GVE      DSECT
000000 GVENXTA DS   A          ADDRESS OF NEXT GVE FOR GCE
000004 GVEFVALN DS   H          LENGTH OF FIRST VALUE
000006 GVESVALN DS   H          LENGTH OF SECOND VALUE
000008 GVEVALS DS   0F          START OF VALUE(S)
00008  GVELEN   EQU  ((*-GVE+3)/4)*4    LENGTH OF GVE IN BYTES
00002  GVELENF  EQU  ((*-GVE+3)/4)     LENGTH OF GVE IN WORDS
```

1.69 #GRLDS

```
COPY #GRLDS
*****
***          GRL: GENERATOR PROCESS RECORD LIST
***          ***
***      GRL IS A DSECT THAT DEFINES THE ELEMENTS ON THE CHAIN
***      DESCRIBING RECORDS USED BY A PROCESS DURING ADS/ONLINE
***      GENERATION.
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------|
| 000000 | GRL | DSECT |
| 000000 | GRLNXTA | DS A |
| 000004 | GRLPREVA | DS A |
| 000008 | GRLGRDA | DS A |
| 00003 | GRLLENF | EQU ((*-GRL+3)/4) |
| 0000C | GRLLEN | EQU GRLLENF*4 |

1.70 #GSADS

```

COPY #GSADS
*****
***      GSA: GENERATOR STORAGE AREA (ADS/ONLINE)
***      GSA IS A DSECT THAT DESCRIBES THE WORK STORAGE AREAS USED
***      BY THE ADS/ONLINE GENERATOR. THESE AREAS INCLUDE: THE DC
***      MACRO PARM AREA, THE SUSBSHEMA CONTROL AREA, MAP AND DATA
***      BASE RECORD BUFFERS, AND AN INTERNAL STACK.
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|---|
| 000000 | GSA DSECT | |
| 000000 | GSAID DS CL4 | 03/30/90 17:01:09 03/18/92 |
| 000004 | SYSPLIST DS 10F | "GSA*" DC MACRO PARM AREA |
| 00002C | GSASCSCA DS F | SAVE AREA FOR STACK CHECKING |
| 000030 | GSASTKEA DS A | STACK END ADDRESS |
| 000034 | GSAGWAKA DS A | ADDRESS OF GWAWK |
| 000038 | GSADGBA DS A | ADDRESS OF DGB |
| | * | GSACSCA BELOW AND THE ADSCSCA FIELD SPECIFIED LATER BOTH |
| | * | CONTAIN THE SAME ADDRESS AND SHOULD BE INITIALIZED TOGETHER |
| 00003C | GSACSCA DS A | ADDRESS OF CSA FOR EVAL CALLS |
| 000040 | GSAFREEA DS A | STORAGE ADDRESS FOR FRESTG |
| 000044 | GSAPRSEA DS A | PRIOR RSE FOR COMPILE |
| 000048 | GSANCMPA DS A | NEXT AVAILABLE BYTE IN FDB |
| 00004C | GSACOMPA DS A | PROCESS TO COMPILE FOR GNRP |
| 000050 | GSAAATBLA DS A | ADDR OF CONVERSION TABLE |
| 000054 | GSAFDBA DS A | ADDR OF FDB FOR DIALOG |
| 000058 | GSAFDBSZ DS F | SIZE OF FDB |
| 00005C | GSAPBUFA DS A | ADDR OF STG FOR SCRATCH READS |
| 000060 | GSANXTBA DS A | ADDR OF NEXT ENTRY IN BUFFER |
| 000064 | GSABFNDA DS A | ADDR OF END OF BUFFER |
| 000068 | GSAPRDEA DS A | ADDR OF LAST RDE POINTER |
| 00006C | GSAQRECA DS A | ADDR OF QUEUE RECORD STORAGE |
| 000070 | GSACPROA DS A | ADDR OF CURRENT GPM/GRS |
| 000074 | GSAQRID DS F | QUEUE RECORD ID |
| 000078 | GSAQSIZE DS F | SIZE OF QUEUE RECORD STORAGE |
| 00007C | GSAAATBLN DS H | # ENTRIES IN ADDRESS TABLE |
| 00007E | GSAFDBP# DS H | # OF 512 PAGES IN FDB |
| 000080 | GSACMSG# DS H | CURRENT MSG CODE IN SCR BUFFER |
| 000082 | GSACERRS DS H | TOTAL COMPILE ERRORS FOR PROCESS |
| | GSAMOUT #FLAG X'80' | ON WHEN SCREEN MAPPED OUT |
| 000084 | GSAMOUTI DS 0XL1 | |
| 000080 | GSAMOUTM EQU X'80' | |
| | GSAFCFD #FLAG X'40' | ON WHEN FIRST FUNCTION FOUND |
| 000084 | GSAFCFDI DS 0XL1 | |
| 000040 | GSAFCFDM EQU X'40' | |
| | GSAACER #FLAG X'20' | ON WHEN ACTION CODE ERROR |
| 000084 | GSAACERI DS 0XL1 | |
| 000020 | GSAACERM EQU X'20' | |
| | GSAPERR #FLAG X'10' | ON WHEN PROCESSING ERROR |
| 000084 | GSAPERI DS 0XL1 | |
| 00010 | GSAPERRM EQU X'10' | |
| | GSAFERR #FLAG X'08' | ON WHEN FUNCTION SELECT ERROR |
| 000084 | GSAFERI DS 0XL1 | |
| 000008 | GSAFERRM EQU X'08' | |
| | GSADPFC #FLAG X'04' | DUPLICATE FUNCTION CODES SELECTED |
| 000084 | GSADPFCI DS 0XL1 | |
| 000004 | GSADPFCM EQU X'04' | |
| | GSAPDEL #FLAG X'02' | ON WHEN PROCESS IS TO BE DELETED |
| 000084 | GSAPDELI DS 0XL1 | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------|-----------------------------------|
| 00002 | GSAPDELM EQU X'02' | |
| | GSAPMOD #FLAG X'01' | |
| 00004 | GSAPMODI DS OXL1 | ON WHEN PROCESS HAS BEEN MODIFIED |
| 00001 | GSAPMODM EQU X'01' | |
| 00004 | GSASFLG1 DS X | FIRST SCREEN FLAG BYTE |
| | GSAPMAP #FLAG X'80' | |
| 00005 | GSAPMAPI DS OXL1 | PREMAP SELECTED |
| 000080 | GSAPMAPM EQU X'80' | |
| | GSANRSP #FLAG X'40' | |
| 000085 | GSANRSP1 DS OXL1 | NEW RESPONSE SELECTED |
| 000040 | GSANRSPM EQU X'40' | |
| | GSARESP #FLAG X'20' | |
| 000085 | GSARESPI DS OXL1 | RESPONSE SELECTED |
| 000020 | GSARESPM EQU X'20' | |
| | GSANCOPY #FLAG X'10' | |
| 000085 | GSANCOPY1 DS OXL1 | NEW COPY RECORD SELECTED |
| 000010 | GSANCOPYM EQU X'10' | |
| | GSADMAP #FLAG X'08' | |
| 000085 | GSADMAP1 DS OXL1 | DISPLAY MAP SELECTED |
| 000008 | GSADMAPM EQU X'08' | |
| | GSAUUMAP #FLAG X'04' | |
| 000085 | GSAUUMAPI DS OXL1 | UPDATE MAP SELECTED |
| 000004 | GSAUUMPM EQU X'04' | |
| | GSADISP #FLAG X'02' | |
| 000085 | GSADISPI DS OXL1 | DISPLAY SELECTED |
| 000002 | GSADISPM EQU X'02' | |
| | GSADIAL #FLAG X'01' | |
| 000085 | GSADIALI DS OXL1 | DIALOG SELECTED |
| 000001 | GSADIALM EQU X'01' | |
| 000085 | GSASFLG2 DS X | SECOND FLAG BYTE |
| | GSAUPRO #FLAG X'80' | |
| 000086 | GSAUPROI DS OXL1 | UPDATE PROCESS SELECTED |
| 000080 | GSAPRNTM EQU X'80' | |
| | GSAPRNT #FLAG X'40' | |
| 000086 | GSAPRNTI DS OXL1 | PRINT SELECTED |
| 000040 | GSAPRNTM EQU X'40' | |
| | GSADREC #FLAG X'20' | |
| 000086 | GSADRECI DS OXL1 | DISPLAY RECORD SELECTED |
| 000020 | GSADRECM EQU X'20' | |
| | GSACOMP #FLAG X'10' | |
| 000086 | GSACOMPI DS OXL1 | COMPILE SELECTED |
| 000010 | GSACOMPM EQU X'10' | |
| | GSQUIT #FLAG X'08' | |
| 000086 | GSQUITI DS OXL1 | QUIT SELECTED |
| 000008 | GSQUITM EQU X'08' | |
| | GSASUSP #FLAG X'04' | |
| 000086 | GSASUSPI DS OXL1 | SUSPEND SELECTED |
| 000004 | GSASUSPM EQU X'04' | |
| | GSAFDOP #FLAG X'02' | |
| 000086 | GSAFDOP1 DS OXL1 | DIALOG OPTIONS SELECTED |
| 000002 | GSAFDOPM EQU X'02' | |
| | GSASWCH #FLAG X'01' | |
| 000086 | GSASWCHI DS OXL1 | SWITCH TASK SELECTED (UMBRELLA) |
| 000001 | GSASWCHM EQU X'01' | |
| 000086 | GSASFLG3 DS X | THIRD SCREEN FLAG BYTE |
| | GSARFSH #FLAG X'80' | |
| 000087 | GSARFSHI DS OXL1 | REFRESH SELECTED |
| 000080 | GSARFSHM EQU X'80' | |
| | GSAPGFD #FLAG X'40' | |
| 000087 | GSAPGFDI DS OXL1 | PAGE FORWARD SELECTED |
| 000040 | GSAPGFDM EQU X'40' | |
| | GSAPGBK #FLAG X'20' | |
| 000087 | GSAPGBK1 DS OXL1 | PAGE BACK SELECTED |
| 000020 | GSAPGBK2 EQU X'20' | |
| | GSACLER #FLAG X'10' | |
| 000087 | GSACLER1 DS OXL1 | CLEAR KEY HIT |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------|---|
| | 00010 GSACLERM EQU X'10' | |
| | GSARETN #FLAG X'10' | REDEFINES RETURN KEY HIT (TOOLBOX) *MET87343* |
| 000087 | GSARETNI DS 0XL1 | |
| | 00010 GSARETNM EQU X'10' | |
| | GSAEINTR #FLAG X'08' | ENTER ONLY HIT |
| 000087 | GSAEINTR DS 0XL1 | |
| | 00008 GSAENTRM EQU X'08' | |
| | GSAEDIT #FLAG X'04' | ON WHEN AUTOMATIC EDIT ERROR |
| 000087 | GSAEDITI DS 0XL1 | |
| | 00004 GSAEDITM EQU X'04' | |
| | GSAMODS #FLAG X'02' | MODULE SCREEN SELECTED *MET87343* |
| 000087 | GSAMODSI DS 0XL1 | |
| | 00002 GSAMODSM EQU X'02' | |
| | GSASRDE #FLAG X'01' | SPECIAL RDE IS BEING PROCESSED |
| 000087 | GSASRDEI DS 0XL1 | |
| | 00001 GSASRDEM EQU X'01' | |
| 000087 | GSASFLG4 DS X | FOURTH FLAG BYTE |
| 000088 | DBL DS D | GENERALLY USEFUL DOUBLE WORD |
| 000090 | FWD DS F | USEFUL FULLWORD |
| 000094 | HALF DS H | GENERALLY USEFUL HALFWORD |
| 000098 | DS OF | |
| | *** GNRC VARIABLE AREA | |
| 000098 | GSAGRDA DS A | CURRENT GRD ADDRESS |
| 00009C | GSACGREA DS A | CURRENT GRE ADDRESS |
| 0000A0 | GSAGCEA DS A | CURRENT GCE ADDRESS |
| 0000A4 | GSAPGREA DS A | ADDR OF PRIOR GRE |
| 0000A8 | GSACINTA DS A | CURRENT INIT BUFFER POSITION |
| 0000AC | GSAOCGRE DS A | HIGHER OCCURS LEVEL GRE |
| 0000B0 | GSALGRE DS A | LAST OCCURRING GRE |
| 0000B4 | GSAREGSV DS F | REGISTER SAVE AREA |
| 0000B8 | GSAOCLVL DS H | CURRENT OCCURS LEVEL |
| | GSAMAPR #FLAG X'80' | ON IF MAP RECORDS PROCESSED |
| 0000BA | GSAMAPRI DS 0XL1 | |
| | 00080 GSAMAPRM EQU X'80' | |
| | GSASDE #FLAG X'40' | ON IF SDES-044 FOR ELEMENT |
| 0000BA | GSASDEI DS 0XL1 | |
| | 00040 GSASDEM EQU X'40' | |
| | GSAEFND #FLAG X'20' | ON IF ELEMENT FOUND |
| 0000BA | GSAEFNDI DS 0XL1 | |
| | 00020 GSAEFNDM EQU X'20' | |
| | GSAOGRD #FLAG X'10' | ON IF GRD FOR RECORD |
| 0000BA | GSAOGRDI DS 0XL1 | |
| | 00010 GSAOGRDM EQU X'10' | |
| | GSASRCD #FLAG X'08' | ON IF SUBSCHEMA RECORD |
| 0000BA | GSASRCDI DS 0XL1 | |
| | 00008 GSASRCDM EQU X'08' | |
| | GSAMFLD #FLAG X'04' | ON IF MAP AND SS REC DIFFERENT |
| 0000BA | GSAMFLDI DS 0XL1 | |
| | 00004 GSAMFLDM EQU X'04' | |
| | GSASVAL #FLAG X'02' | ON IF INIT 2ND VALUE FOR GCE |
| 0000BA | GSASVALI DS 0XL1 | |
| | 00002 GSASVALM EQU X'02' | |
| | GSAPWA #FLAG X'01' | ON IF STORAGE REQUEST FOR PWA |
| 0000BA | GSAPWAI DS 0XL1 | |
| | 00001 GSAPWAM EQU X'01' | |
| 0000BA | GNRCFLG DS X | FLAG BYTE FOR GNRC |
| | GSAFUNC #FLAG X'80' | ON IF FUNCTION CODE FOR GRS |
| 0000BB | GSAFUNCI DS 0XL1 | |
| | 00080 GSAFUNCM EQU X'80' | |
| | GSACBCE #FLAG X'40' | ON IF BCE HAS BEEN CHANGED |
| 0000BB | GSACBCEI DS 0XL1 | |
| | 00040 GSACBCEM EQU X'40' | |
| | GSACPFK #FLAG X'20' | ON IF CONTROL KEY HAS BEEN CHANGED |
| 0000BB | GSACPFKI DS 0XL1 | |
| | 00020 GSACPFKM EQU X'20' | |
| | GSASQSN #FLAG X'10' | ON IF SIGNED ON RCM COMPILER *LMA87321* |

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|--|
| 0000BB | GSASQSNI DS OXL1 | |
| 00010 | GSASQSNM EQU X'10' | |
| | *SAGNRT #FLAG X'08' | ON IF PROCESSING A GENERATE REQUEST. SET IN ADSGNP1. |
| | * | ADS/12 |
| | GSAOFMT #FLAG X'04' | ON IF IDMS/PC-STYLE (OSCAR) FORMAT |
| 0000BB | GSAOFMTI DS OXL1 | |
| 00004 | GSAOFMTM EQU X'04' | |
| | GSAAFMT #FLAG X'02' | ON IF ADS+/PC-STYLE (APPC) FORMAT |
| 0000BB | GSAAFMTI DS OXL1 | |
| 00002 | GSAAFMTM EQU X'02' | |
| 0000BB | GNRFLAG DS X | |
| | GSASCMD #FLAG X'80' | ON WHEN SCHEMA/SUBSCH MODIFIED |
| 0000BC | GSASCMDI DS OXL1 | |
| 00080 | GSASCMDM EQU X'80' | |
| | GSAMPMD #FLAG X'40' | ON WHEN MAP MODIFIED |
| 0000BC | GSAMPMDI DS OXL1 | |
| 00040 | GSAMPMDM EQU X'40' | |
| 0000BC | GNDGFLAG DS X | |
| | GSALINE #FLAG X'80' | ON WHEN GNDS CALL IS FOR LINEOUTS |
| 0000BD | GSALINEI DS OXL1 | |
| 00080 | GSALINEM EQU X'80' | |
| | GSAFIRS #FLAG X'40' | GENERALLY USEFUL FIRST-TIME SWITCH |
| 0000BD | GSAFIRSI DS OXL1 | |
| 00040 | GSAFIRSM EQU X'40' | |
| | GSANCMT #FLAG X'20' | NEW COPY ENTRIES EMPTY ON SCREEN |
| 0000BD | GSANCMTI DS OXL1 | |
| 00020 | GSANCMTM EQU X'20' | |
| 0000BD | GNDFLAG DS X | |
| | GSASORT #FLAG X'80' | ON WHEN NEED TO CHECK SORTED SET |
| 0000BE | GSASORTI DS OXL1 | |
| 00080 | GSASORTM EQU X'80' | |
| | GSASPFI #FLAG X'40' | ON WHEN NEED TO CHECK SPF SET |
| 0000BE | GSASPFI DS OXL1 | |
| 00040 | GSASPFM EQU X'40' | |
| | GSABIND #FLAG X'08' | ON WHEN RUN-UNIT BOUND |
| 0000BE | GSABINDI DS OXL1 | |
| 00008 | GSABINDM EQU X'08' | |
| 0000BE | GNDFLAG DS X | |
| | GSALRE #FLAG X'08' | ON IF DN QUALIFIED BY LRE *GWG90088 |
| 0000BF | GSALREI DS OXL1 | |
| 00008 | GSALREM EQU X'08' | |
| | GSAEDUP #FLAG X'04' | ON IF DUPLICATE ERROR *GWG99147 |
| 0000BF | GSAEDUPI DS OXL1 | |
| 00004 | GSAEDUPM EQU X'04' | |
| | GSALRCD #FLAG X'80' | ON IF LR BEING PROCESSED |
| 0000BF | GSALRCDI DS OXL1 | |
| 00080 | GSALRCDM EQU X'80' | |
| | GSAROLE #FLAG X'40' | ON IF RECORD NAME = ROLE NAME |
| 0000BF | GSAROLEI DS OXL1 | |
| 00040 | GSAROLEM EQU X'40' | |
| | GSALRWK #FLAG X'20' | ON IF LR PARTICIPANT IS WORK REC |
| 0000BF | GSALRWKI DS OXL1 | |
| 00020 | GSALRWKM EQU X'20' | |
| | GSAPBLD #FLAG X'10' | ON IF BUILDING GRD FOR PRIMARY REC |
| 0000BF | GSAPBLDI DS OXL1 | |
| 00010 | GSAPBLDM EQU X'10' | |
| | LRCDFLAG DS X | LOGICAL RECORD FLAGS |
| 0000C0 | DS 0F | |
| 0000C0 | GNDBBUF DS CL64 | BUFFER FOR GNDB & SOME GENR CALLS |
| 000100 | GNRCBUF DS CL64 | BUFFER FOR GNRC CALLS |
| 000140 | GSALNBUF DS CL80 | LINEOUT BUFFER FOR GNDS |
| 000190 | DS 0F | |
| 000190 | SDRBUF DS CL108 | SDR-042 BUFFER |
| 0001FC | SDESBUF DS CL108 | SDES-044 BUFFER |
| 000268 | RECBUFF DS CL84 | BUFFER FOR RECSYN RECORD |
| 0002BC | DBBUFF DS CL402 | GENERAL BUFFER FOR GNDB |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--|
| 00044E | | ORG SDRBUF |
| 000190 | SYSBUF | DS CL512 * BUFFER FOR SYS-041 REC |
| 000390 | | ORG SDRBUF |
| 000190 | CVGBUF | DS CL320 * BUFFER FOR CVGDEF REC |
| 0002D0 | | ORG SDRBUF |
| 000190 | SYMBUF | DS CL512 BUFFER FOR SYMTEXT AND SYMCTL |
| 000390 | | ORG |
| 000450 | RECDBKEY | DS F DBKEY AREA FOR ACCEPTS |
| 000454 | MAPRDBKY | DS F DBKEY FOR MAP RECORD |
| 000458 | MP125KEY | DS F DBKEY OF MAPRCD-125 |
| 00045C | GSAIMDBK | DS F BATCH INPUT MAP DBKEY *JMA85070* |
| 000460 | GSAOMDBK | DS F BATCH OUTPUT MAP DBKEY *JMA85070* |
| 000464 | GSAFNGRD | DS A ADDR OF GRD MATCH IN GNRC |
| 000468 | GSAFNGRE | DS A ADDR OF GRE MATCH IN GNRC |
| 00046C | GSAGLRA | DS A ADDR OF CURRENT GLR |
| 000470 | HOLDVER | DS H HOLD AREA FOR VERSION NUMBERS |
| 000472 | GSADRLVL | DS CL2 HOLD AREA FOR RECORD LEVEL |
| 000474 | GSARELSB | DS CL1 RELATIVE SELECT BYTE FOR FUNC'D |
| 000475 | GSASCHAR | DS CL1 MODIFIED SELECT BYTE CHARACTER |
| 000478 | GSARGSV2 | DS F SECONDARY REGISTER SAVE WORD |
| 00047C | GSARCPCH | DS A ADDR OF ADSOGNRC PATCH AREA |
| 000480 | GSADBPCH | DS A ADDR OF ADSOGNDB PATCH AREA |
| 000484 | GSANCSAV | DS F REGISTER SAVE AREA FOR GNNC |
| 000488 | GSANCSUB | DS H SUBSCRIPT SAVE AREA FOR GNNC |
| 00048A | | DS H REGISTER SAVE AREA FOR GNNC |
| 00048C | GSAPSAA | DS A ADDR OF PARSER STORAGE AREA |
| 000490 | GSAINTLN | DS H FIELD INIT LENGTH FOR GNRI CALL |
| 000492 | GSANCLN | DS H NEW COPY LINE # VALUE |
| 000494 | GSAAPLA | DS A * ADDR OF ADSOAUINI-MAP RECORD |
| 000498 | GSAAWKA | DS A * ADDR OF ADSOAWRK RECORD |
| 00049C | GSAADBA | DS A * ADDR OF ADB |
| 0004A0 | GSAAPRMA | DS A * ADDR OF PARMS PASSED TO AGENT |
| 0004A4 | GSANAGRC | DS H * NUMBER OF APPL GLOBAL RECS |
| 0004A6 | GSANTSKS | DS H * NUMBER OF GTCS *MDR05086* |
| 0004A8 | GSAMDNL | #FLAG X'80' |
| | GSAMDNL1 | DS 0XL1 * MODIFY BUT NO LOAD MODULE EXISTS |
| | 00080 | GSAMDNL M EQU X'80' |
| | * | ALSO USED BY ADSG FOR SEC CHECK |
| 0004A8 | | GSAGENR #FLAG X'40' |
| | | DS 0XL1 * GENERAL SYSCMT RECORD READ |
| 0004A8 | 00040 | GSAGENRM EQU X'40' |
| | | GSASECR #FLAG X'20' |
| 0004A8 | 00020 | GSASECRI DS 0XL1 * SECURITY SYSCMT RECORD READ |
| | | GSASECRM EQU X'20' |
| 0004A8 | 00010 | GSAERR #FLAG X'10' |
| | | DS 0XL1 * ON IF TERMINAL ERROR OCCURRED |
| 0004A8 | 00010 | GSAEERRI DS 0XL1 |
| | | GSAEERRM EQU X'10' |
| 0004A8 | 00008 | GSAGNLF #FLAG X'08' |
| | | DS 0XL1 * ON IF GENERAL REC FOR FUNCTION |
| 0004A8 | 00008 | GSAGNLFI DS 0XL1 |
| | | GSAGNLFM EQU X'08' |
| 0004A8 | 00004 | GSAACFG #FLAG X'04' |
| | | DS 0XL1 * ON IF APPL GENERATOR IS RUNNING |
| 0004A8 | 00004 | GSAACFGI DS 0XL1 |
| | | GSAACFGM EQU X'04' |
| 0004A8 | 00002 | GSANGTC #FLAG X'02' |
| | | DS 0XL1 * ON IF NO MORE GTC'S TO GO TO TAT |
| 0004A8 | 00002 | GSANGTCI DS 0XL1 |
| | | GSANGTCM EQU X'02' |
| 0004A8 | 00001 | GSADTAT #FLAG X'01' |
| | | DS 0XL1 * ON IF TAT IS DELETED IN ENTIRETY |
| 0004A8 | 00001 | GSADTATI DS 0XL1 |
| | | GSADTATM EQU X'01' |
| 0004A8 | | GSAAPFLG DS X |
| | | GSADUP #FLAG X'80' |
| 0004A9 | | GSADUPI DS 0XL1 |
| | | GSADUPM EQU X'80' |
| 0004A9 | 00080 | GSAMBND #FLAG X'40' |
| | | DS 0XL1 ON IF DDLCMSG AREA IS BOUND |
| 0004A9 | 00040 | GSAMBNDI DS 0XL1 |
| | | GSAMBNDM EQU X'40' |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------|---------------------------------------|
| 0004A9 | GSABTCH #FLAG X'20' | ON IF RUNNING IN BATCH MODE |
| 00020 | GSABTCHI DS OXL1 | |
| 0004A9 | 00010 GSABTCHM EQU X'20' | ON IF BIND ERROR -- GNDB FUNC#1 |
| 0004A9 | GSALDERI DS OXL1 | |
| 0004A9 | GSALDERM EQU X'10' | |
| 0004A9 | GSASBND #FLAG X'08' | ON IF MINIMUM SUBSCHEMA BIND DONE |
| 0004A9 | GSASBNDI DS OXL1 | |
| 00008 | GSASBNDM EQU X'08' | |
| 0004A9 | GSASABT #FLAG X'04' | ON IF SUBTASK ABORTED (ASF & UMBR) |
| 0004A9 | GSASABTI DS OXL1 | |
| 00004 | GSASABTM EQU X'04' | |
| 0004A9 | GSAAABT1 #FLAG X'02' | ON IF 1ST SUBTASK ABORT (ASF/UMBR) |
| 0004A9 | GSAAABT1I DS OXL1 | |
| 00002 | GSAAABT1M EQU X'02' | |
| 0004A9 | GSARGEN #FLAG X'01' | ON IF ADSORGREN IS EXECUTING |
| 0004A9 | GSARGENI DS OXL1 | |
| 00001 | GSARGENM EQU X'01' | |
| 0004A9 | BGENFLG DS X | BATCH GENERATOR & SUBTASK FLAG BYT |
| 0004AA | GSAAUSF DS OXL1 | AUTHORIZED USER FOUND |
| 00080 | GSAAUSFM EQU X'80' | |
| 0004AA | GSAAUSR #FLAG X'40' | USER-047 RECORD FOUND |
| 0004AA | GSAAUSRFI DS OXL1 | |
| 00040 | GSAAUSRFM EQU X'40' | |
| 0004AA | GSANOLD #FLAG X'20' | NO FDB LOAD MODULE FOUND |
| 0004AA | GSANOLDI DS OXL1 | |
| 00020 | GSANOLDM EQU X'20' | |
| 0004AA | GSARGFL DS X | REGISTRATION/SECURITY FLAG BYTE |
| 0004AA | GSAOFCBM #FLAG X'80' | ON IF MAIN OCB CVGDEF REC READ |
| 0004AB | GSAOFCBMI DS OXL1 | |
| 00080 | GSAOFCBMM EQU X'80' | |
| 0004AB | GSAOFCBS #FLAG X'40' | ON IF STAT DEF REC OCB CVGDEF REC |
| 0004AB | GSAOFCBSI DS OXL1 | |
| 00040 | GSAOFCBSM EQU X'40' | |
| 0004AB | GSAUFNC #FLAG X'20' | ** PARSING A USER FUNCTION *LCB84137* |
| 0004AB | GSAUFNCI DS OXL1 | |
| 00020 | GSAUFNCM EQU X'20' | |
| 0004AB | *SANNXT #FLAG X'10' | NO XDENEXT FIELD GENERATED*MCM87100* |
| 0004AB | GSABSFL DS X | ADSOBSYS FLAG BYTE |
| 0004AC | SYSDBKEY DS F | * DBKEY OF SYS-041 RECORD |
| 0004B0 | GSAASFNV DS OF | * ADSA CALL PARM SAVE *MDR05086* |
| 0004B0 | GSAASFN2 DS H | * LEVEL 2 FUNCTION CODE *MDR85086* |
| 0004B2 | GSAAGFOP DS X | * OPERATION CODE *MDR85086* |
| 0004B3 | GSAAGAFO DS X | * OCCURRENCE CODE *MDR85086* |
| 0004B4 | GSAGARIX DS H | * CURRENT HIGH RESPONSE INDEX |
| 0004B6 | GSARKEIX DS H | * CURRENT HIGH RESP FUNC KEY INDEX |
| 0004B8 | GSAAQRET DS H | * ADSA Q RETENTION PERIOD *MDR85086* |
| 0004BA | GSAUTFNC DS H | * # OF UNDEF TSK TOP FUNCS*MDR85086* |
| 0004BC | GSAFCNIX DS H | * CURRENT HIGH FUNCTION INDEX |
| 0004BE | GSADUPTK DS H | * COUNT OF DUPLICATE TASK CODES |
| 0004C0 | GSAURCNT DS H | * # OF USING REC NAMES FOR FCN |
| 0004C2 | GSACMTEO DS H | * CURRENT MENU TABLE OFFSET |
| 0004C4 | GSAURLIT DS F | * ADDR OF USING REC LITERAL BUILD |
| 0004C8 | GSACARDA DS A | * CURRENT ARD ADDRESS. *MDR85086* |
| 0004CC | GSALPOFF DS F | * NEXT AVAILABLE LITPOOL OFFSET |
| 0004D0 | GSACRBMA DS A | * CURRENT RESP BIT MAP ADDR |
| 0004D4 | GSACKBMA DS A | * CURRENT KEY BIT MAP ADDR |
| 0004D8 | GSARBMLF DS H | * RESPONSE BIT MAP # FULL WORDS |
| 0004DA | GSAKBMLF DS H | * KEY BIT MAP # FULL WORDS |
| 0004DC | GSAKBMLN DS H | * KEY BIT MAP # BITS |
| 0004DE | GSAADBP# DS H | * NUMBER OF PAGES FOR ADB LOAD MOD |
| 0004E0 | GSABPOFF DS F | * NEXT AVAIL OFFSET IN BITMAP POOL |
| 0004E4 | GSAADBSZ DS F | * SIZE OF ADB UNDER CONSTRUCTION |
| 0004E8 | GSATATA DS A | * ADDR OF TASK APPLICATION TABLE |
| 0004EC | GSATATSZ DS F | * SIZE OF TASK APPLICATION TABLE |

| Offset | Value | |
|--|---------------------|---|
| 0004F0 | GSANTATA DS A | * NEW TASK APPL TABLE ADDR |
| 0004F4 | GSANTATS DS F | * NEW TASK APPL TABLE SIZE |
| 0004F8 | GSANDELT DS H | * # OF TASKS TO DELETE FROM TAT |
| 0004FA | GSANMODT DS H | * # OF TASKS IN MODIFIED TAT |
| 0004FC | GSAOCBA DS A | ADDR OF OCB BUILT BY ADSOBSYS |
| 000500 | GSAMSQCT DS H | * # OF SEQUENCED RESPS FOR FUNC |
| 000502 | GSACNMSQ DS H | * # OF MSQ'S FOR CURRENT FUNCTION |
| 000504 | GSATATP# DS H | * # OF PAGES FOR TAT LOAD MODULE |
| 000506 | ORG GSAASFV | **SQL** |
| <hr/> | | |
| *****SQL*** | | |
| * THE FOLLOWING FIELDS ARE FOR ADS GENERATOR AND ARE | | **SQL** |
| * REDEFINED ABOVE FOR ADSA. | | **SQL** |
| <hr/> | | |
| 0004B0 | DS OF | **SQL** |
| | GSASQL #FLAG X'80' | 1ST TIME SQL FLAG FOR PARSING**SQL** |
| 0004B0 | GSASQLI DS 0XL1 | |
| 00080 | GSASQLM EQU X'80' | |
| | GSASQWW #FLAG X'40' | WHENEVER WARNING PARSED **SQL** |
| 0004B0 | GSASQWMI DS 0XL1 | |
| 00040 | GSASQWWM EQU X'40' | |
| | GSASQWE #FLAG X'20' | WHENEVER ERROR PARSED **SQL** |
| 0004B0 | GSASQWEI DS 0XL1 | |
| 00020 | GSASQWEM EQU X'20' | |
| | GSASQWN #FLAG X'10' | WHENEVER NOT FOUND PARSED **SQL** |
| 0004B0 | GSASQWNI DS 0XL1 | |
| 00010 | GSASQWNM EQU X'10' | |
| | GSASQLE #FLAG X'08' | SQL ERROR **SQL** |
| 0004B0 | GSASQLEI DS 0XL1 | |
| 00008 | GSASQLEM EQU X'08' | |
| | GSASQFL1 DS X | FLAG BYTE ONE FOR SQL **SQL** |
| | GSASQCN #FLAG X'80' | SQLXQ CONNECT ISSUED **SQL** |
| 0004B1 | GSASQCNI DS 0XL1 | |
| 00080 | GSASQCNM EQU X'80' | |
| | GSASQSC #FLAG X'40' | SQLXQ SET COMPILES ISSUED *JMA91091* |
| 0004B1 | GSASQSCI DS 0XL1 | |
| 00040 | GSASQSCM EQU X'40' | |
| | GSASQLF2 DS X | FLAG BYTE TWO FOR SQL **SQL** |
| 0004B2 | GSASQLCT DS H | SQL STATEMENTS COUNTER **SQL** |
| 0004B4 | GSASQLWA DS F | SQL WHENEVER BLOCK ADDRESS **SQL** |
| 0004B8 | GSASQXAL DS A | A(SQLXQ ARGUMENT INDEX LIST) **SQL** |
| 0004BC | GSASQHGP DS A | SQL HOST VAR GROUP HVD ENTRY **SQL** |
| 0004C0 | GSASQHVT DS A | SQL HOST VAR TBL CHAIN ANCHOR**SQL** |
| 0004C4 | GSASQIPM DS A | SQLIS PARM LIST ADDRESS **SQL** |
| 0004C8 | GSASQKPM DS F | KEY FOR CURRENT SQLIS PARM LST**SQL** |
| 0004CC | DS F | DON'T USE(REFDEFINES GSALPOFF) JMA90347 |
| 0004D0 | GSASQIPC DS A | SQLIS PARM LIST CHAIN **SQL** |
| 0004D4 | GSASQRPB DS A | A(SQLRPB) FOR SQLIS/SQLXQ **SQL** |
| 0004D8 | GSASQSSI DS A | A(SQLSSI) FOR SQLIS/SQLXQ **SQL** |
| 0004DC | GSASQCA DS A | A(SQLCA) FOR SQLIS/SQLXQ **SQL** |
| 0004E0 | GSASQPIB DS A | A(SQLPIB) FOR SQLIS CALL **SQL** |
| 0004E4 | GSASQPBX DS A | A(SQLPIB) FOR SQLXQ CALL **SQL** |
| 0004E8 | GSASQCIB DS A | A(SQLCIB) FOR SQLXQ CALL **SQL** |
| 0004EC | GSASQLIN DS A | A(WORKING LINE) FOR ADSOSQL **SQL** |
| 0004F0 | GSASQHVC DS H | # ENTRIES IN HOST VAR TBL CHN**SQL** |
| 0004F2 | GSASQLRC DS H | # SQL REQUIRED RECS PROCESSD **SQL** |
| 0004F4 | GSASQLTO DS H | SQL SAVE TOKEN OFFSET **SQL** |
| 0004F6 | GSASQHGC DS H | HOST VAR COUNT OF GRP ELEMS **SQL** |
| 0004F8 | GSASQHGR DS CL2 | HOST VAR GROUP RECORD LEVEL **SQL** |
| 0004FA | DS CL2 | RESERVED **SQL** |
| 0004FC | GSASQLHV DS F | COUNTER FOR ADSOGRNS **SQL** |
| 000500 | GSASQXPM DS A | SQLXQ PARM LIST ADDRESS **SQL** |
| 000504 | DS CL2 | RESERVED **SQL** |
| 000506 | ORG , | **SQL** |
| | GSAPREV #FLAG X'80' | PREV STEP *MET87343* |
| 000506 | GSAPREVI DS 0XL1 | |
| 00080 | GSAPREVM EQU X'80' | |

| <u>Offset</u> | <u>Value</u> | | | |
|---|----------------------|--------------------------------------|--|------------|
| 000506 | GSANEXT #FLAG X'40' | NEXT STEP | | *MET87343* |
| | GSANEXTI DS 0XL1 | | | |
| 00040 | GSANEXTM EQU X'40' | | | |
| | GSAPGTO #FLAG X'20' | PAGE-TO SELECTED | | *MET87343* |
| 000506 | GSAPGTOI DS 0XL1 | | | |
| 00020 | GSAPGTM EQU X'20' | | | |
| | GSACERR #FLAG X'10' | GLOBAL COMPILE ERROR | | *MET87343* |
| 000506 | GSACERRI DS 0XL1 | | | |
| 00010 | GSACERRM EQU X'10' | | | |
| | GSACTRP #FLAG X'08' | GLOBAL COMPILE LOOP TRAP | | *MET87343* |
| 000506 | GSACTRPI DS 0XL1 | | | |
| 00008 | GSACTRPM EQU X'08' | | | |
| | GSAGEN2 #FLAG X'04' | CALL GEN2 (NOT JUST EDIT) | | *MET87343* |
| 000506 | GSAGEN2I DS 0XL1 | | | |
| 00004 | GSAGEN2M EQU X'04' | | | |
| | #FLAG X'02' | | | |
| | #FLAG X'01' | | | |
| 000506 | GSATBFLG DS X | FLAG BYTE FOR TOOL BOX ADSG | | |
| | GSADAWA #FLAG X'80' | DWA WORK AREA FOR DNB BUILDING | | |
| 000507 | GSADWAI DS 0XL1 | | | |
| 00080 | GSADWAM EQU X'80' | | | |
| | GSASHRT #FLAG X'40' | ALLOCATE SHORT STORAGE FOR GWA | | |
| 000507 | GSASHRTI DS 0XL1 | | | |
| 00040 | GSASHRTM EQU X'40' | | | |
| | GSAOSSRE #FLAG X'20' | OLD SRE FRMT RECOVERD FROM SUSPEND | | |
| 000507 | GSAOSSREI DS 0XL1 | | | |
| 00020 | GSAOSSREM EQU X'20' | | | |
| | GSAMBB #FLAG X'10' | MBB FOUND IN OCCURRING GRP*LCB84237* | | |
| 000507 | GSAMBBI DS 0XL1 | | | |
| 00010 | GSAMBBM EQU X'10' | | | |
| | GSANMBB #FLAG X'08' | OTHER THAN MBB IN OCCR GRP*LCB84237* | | |
| 000507 | GSANMBBI DS 0XL1 | | | |
| 00008 | GSANMBBM EQU X'08' | | | |
| | GSAGBLT #FLAG X'04' | GRD BUILT ON FIND REQUEST *JMA90351* | | |
| 000507 | GSAGBLTI DS 0XL1 | | | |
| 00004 | GSAGBLTM EQU X'04' | | | |
| 000507 | GNRCFLG2 DS X | 2ND FLAG BYTE FOR GNRC | | |
| 000508 | GSAWDBK DS F | WORK DBKEY SAVE FIELD | | |
| 00050C | GSAWDBK2 DS F | ANOTHER WORK DBKEY SAVE FIELD | | |
| 000510 | GSAMODNM DS 8F | OCB LOAD MODULE NAME | | |
| 000530 | GSAPASSW DS CL8 | PASSWORD OF USER SIGNED ON | | |
| 000538 | GSAFDNBA DS A | ADDR OF 1ST DIALOG NAME BLOCK | | |
| 00053C | GSACDNBA DS A | ADDR OF CURR DIALOG NAME BLOCK | | |
| ***** | | | | |
| *** THE FOLLOWING FIELDS ARE USED BY ADSA TO SAVE CONTROL BLOCK *** | | | | |
| *** ADDRESSES DURING A PSEUDO-CONVERSE *** | | | | |
| ***** | | | | |
| 000540 | GSACGABA DS A | * CURRENT GAB ADDRESS | | *MDR85086* |
| 000544 | GSACGFEA DS A | * CURRENT GFE ADDRESS | | *MDR85086* |
| 000548 | GSACGARA DS A | * CURRENT GAR ADDRESS | | *MDR85086* |
| 00054C | GSACGFRA DS A | * CURRENT GFR ADDRESS | | *MDR85086* |
| 000550 | GSACMSQA DS A | * CURRENT MSQ ADDRESS | | *MDR85086* |
| 000554 | GSACURNAS DS A | * CURRENT URN ADDRESS | | *MDR85086* |
| 000558 | GSACGTC DS A | * CURRENT GTC ADDRESS | | *MDR85086* |
| 00055C | GSACGMDS DS A | * CURRENT GME ADDRESS | | *MDR85086* |
| ***NOTE: GSACARD - THE CURRENT ARD ADDRESS IS DEFINED EARLIER IN | | | | |
| *** THIS DSECT. IT'S OUT OF SEQUENCE DUE TO LACK | | | | |
| *** OF FULLWORDS HERE. | | | | |
| ***** | | | | |
| *** THE FOLLOWING FIELDS ARE A PARAMETER AREA USED BY THE ADSA *** | | | | |
| *** SESSION JOURNALLING ROUTINE *** | | | | |
| ***** | | | | |
| 000560 | GSAJPCT DS F | * JOURNAL PARAMETER COUNT | | *MDR85086* |
| 000564 | GSAJP1L DS F | * JOURNAL PARM 1 LENGTH | | *MDR85086* |
| 000568 | GSAJP1A DS A | * JOURNAL PARM 1 ADDRESS | | *MDR85086* |
| 00056C | GSAJP2L DS F | * JOURNAL PARM 2 LENGTH | | *MDR85086* |

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|--|
| 000570 | GSAJP2A DS A | * JOURNAL PARM 2 ADDRESS *MDR85086* |
| 000574 | GSAJP3L DS F | * JOURNAL PARM 3 LENGTH *MDR85086* |
| 000578 | GSAJP3A DS A | * JOURNAL PARM 3 ADDRESS *MDR85086* |
| 00057C | ORG GSAJPCT | * |
| 000560 | GSAARAWK DS A | * ADDRESS OF ADSOAWRK *MDR85086* |
| 000564 | GSAARUNI DS A | * ADDRESS OF ADSOAUNI *MDR85086* |
| 000568 | GSAARCPL DS 5A | * ADDITIONAL PARMS *MDR85086* |
| 00057C | GSACGVEA DS A | ADDR OF CURRENT GVE OF GCE |
| 000580 | GSAGRDS DS A | ADDR OF GRD BEING USED (GNNC) |
| 000584 | GSAFUNC DS H | FUNC NUM USED IN ADSOGACT |
| 000586 | GSALENSV DS H | * FOR GNRC - HOLDS FIELD LEN |
| 000588 | GSARGSV3 DS F | 3RD WORD FOR REGISTER SAVES |
| 00058C | GSARACTA DS A | ADDR OF RECORD ACTIVITY CNTL BLK |
| 000590 | GSASACTA DS A | ADDR OF SET ACTIVITY CNTL BLK |
| 000594 | GSARGSV4 DS F | 4TH WORD FOR REGISTER SAVE*LCB84110* |
| 000598 | GSAFDBFS DS F | FDB BUFFER SIZE *LCB84117* |
| 00059C | GSAMSTBA DS A | ** USER FUNC MASTER TBL ADDR *LCB84137* |
| 0005A0 | GSALMTBA DS A | ** USER FUNC XDE MODEL LOAD M*LCB84137* |
| 0005A4 | GSAFNFTBA DS A | ** USER FUNC FUNC TABLE ADDR *LCB84137* |
| 0005A8 | GSACFNTA DS A | ** USER FUNC CURR FUNC TBL AD*LCB84137* |
| 0005AC | GSAFNC#E DS H | ** USER FUNC MAST TBL # ENTRI*LCB84137* |
| 0005AE | GSARDLVL DS H | Used to save Redef level in FDE gen *MCM88086* |
| 0005B0 | GSACXMDA DS A | ** USER FUNC XDE MODEL HDR *LCB84137* |
| 0005B4 | GSAMFTLN DS F | ** USER FUNC MASTER TBL LEN *LCB84138* |
| 0005B8 | GSACPTCH DS A | GEN1,RGEN,BGEN COMMON MODULE PATCH |
| 0005BC | DATEPARM DS 0F | |
| 0005BC | DS A | ADDR OF GREGDATE |
| 0005C0 | DS AL1(128) | |
| 0005C1 | DS AL3(TIME) | |
| 0005C4 | GREGDATE DS CL8 | |
| 0005CC | TIME DS CL8 | |
| 0005D4 | GSAPATCH DS A | PATCH AREA ADDRESS |
| | | * THE FOLLOWING ITEMS ARE USED BY ADSOGNRI THE RECORD INITIALIZATION |
| | | * ROUTINE. |
| | *IXDERST DS 4F | RESULT XDE |
| | *IXDESRC DS 4F | SOURCE XDE |
| 0005D8 | RIXDEADR DS F | ADDRESS OF GETSTG'D AREA FOR BUILDING XDE LIST |
| 0005DC | RIPLIST DS 5F | PLIST FOR EVAL |
| 0005F0 | RIEVALWK DS F | ADDR OF WORK AREA FOR EVAL |
| 0005F4 | DS 6F | RESERVED FOR FUTURE USE |
| 00060C | RIVXDE DS 9F | ROOM FOR 3 VXDE'S |
| 000630 | RIWKDATA DS CL34 | EDITED SOURCE VALUE |
| | RIDECFN #FLAG X'80' | ON IF DEC PT FOUND IN VALUE |
| 000652 | 00080 RIDECFNI DS 0XL1 | |
| | RIDECFNM EQU X'80' | |
| | RISTG #FLAG X'40' | ON IF STORAGE GOTTEN FOR EVAL |
| 000652 | 00040 RISTGI DS 0XL1 | |
| | RISTGM EQU X'40' | |
| | RIFLDLN #FLAG X'20' | ON IF USING L'FIELD NOT L'VALUE |
| 000652 | 00020 RIFLDLNI DS 0XL1 | |
| | RIFLDLNM EQU X'20' | |
| 000652 | RISWITCH DS X | SWITCHS |
| | * END OF ITEMS FOR ADSOGNRI THE RECORD INITIALIZATION ROUTINE | |
| | @SSCTRL | |
| 000658 | DS 0D | |
| 000658 | SSCTRL DS 0CL200 | |
| 000658 4 | PGMNAME DC CL8' ' | |
| 000660 F | ERRSTAT DC C'1400' | |
| 000664 0 | DBKEY DC F'0' | |
| 000668 4 | RECNAMC DC CL16' ' | |
| 000678 4 | AREANAME DC CL16' ' | |
| 000688 4 | ERRORSET DC CL16' ' | |
| 000698 4 | ERRORREC DC CL16' ' | |
| 0006A8 4 | ERRAREA DC CL16' ' | |
| 0006B8 | SSCIDBCM DS 0F | |
| 0006B8 0 | IDBMSCOM DC 25F'0' | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------|---|
| 00071C | ORG | IDBMSCOM |
| 0006B8 | SSCPINFO DS | OF CURRENT PAGE GROUP/DBKEY RADIX |
| 0006B8 0 | SSCPGRUP DC | H'0' CURRENT PAGE GROUP |
| 0006BA 0 | SSCRADIX DC | H'0' CURRENT DBKEY RADIX |
| 0006BC | ORG | , |
| 00071C 0 | DIRDBKEY DC | F'0' |
| 000720 | DBSTATUS DS | OCL7 |
| 000720 4 | DBSTMTCDC | CL2' ' |
| 000722 4 | DBSTATCD DC | CL5' ',CL1' ' |
| 000728 0 | RECOCCUR DC | F'0' |
| 00072C 0 | DMLSEQ DC | F'0' |
| 00730 | GSAINTL EQU | ((*-GSA+3)/4)*4 LENGTH OF GSA TO INITIALIZE |
| 000730 | MRB2PLST DS | 20F MAPPING PLIST AREA |
| | | **** HERE FOLLOWS THE MAPPING WORK AREAS |
| | *DSGMPDG DS | OF DIALOG SCREEN MRB |
| | *DSGMPPM DS | OF PREMAP SCREEN MRB |
| | *DSGMPPRS DS | OF RESPONSE SCREEN MRB |
| | *DSGMPPNC DS | OF NEW COPY SCREEN MRB |
| | *DSGMPDO DS | OF DIALOG OPTIONS SCREEN MRB |
| 000780 | GSAMRB DS | A ADDRESS OF MRB FOR GENERATOR MAPS |
| 000784 | GSAPRMA DS | A ADDR OF PARM LIST (UMBRELLA) |
| 000788 | GSAPROD DS | CL8 PRODUCT NAME FOR UMBRELLA PRODUCTS |
| 000790 | GSAUSES DS | OCL16 SESSION DESCRIPTOR FOR UMBRELLA |
| 000790 | GSAUSDLG DS | CL8 DIALOG NAME |
| 000798 | GSAUSVER DS | CL4 DIALOG VERSION |
| 00079C | DS | CL4 EXTRA BYTES IN SESSION DESCRIPTOR |
| | GSUMBR #FLAG X'80' | ADSG IS RUNNING UNDER THE UMBRELLA |
| 0007A0 | GSUMBRI DS | 0XL1 |
| 00080 | GSUMBMR EQU | X'80' RESUMING OLD UMBRELLA SESSION |
| 0007A0 | GSQAURES #FLAG X'40' | |
| 00040 | GSQAURESI DS | 0XL1 |
| 0007A0 | GSQAURESM EQU | X'40' ON IF RESPONSE PROC IS DUPLICATE |
| 0007A0 | GSARDUP #FLAG X'20' | |
| 00020 | GSARDUPI DS | 0XL1 |
| 0007A0 | GSARDUPM EQU | X'20' ON IF NO DLG NAME ENTERED*RQE84138* |
| 0007A0 | GSANODL #FLAG X'10' | |
| 00010 | GSANODLI DS | 0XL1 |
| 0007A0 | GSANODLM EQU | X'10' ON IF CHECKING FOR DUP DN*GWG90088* |
| 0007A0 | GSAGRE #FLAG X'08' | |
| 00008 | GSAGREI DS | 0XL1 |
| 0007A0 | GSAGREM EQU | X'08' GENERAL PURPOSE FLAG 1 |
| 0007A1 | GSAGPFL1 DS | X RESERVED FOR FUTURE USE |
| | DS | XL3 |
| 0007A4 | GSADGEPI DS | A GDNG ENTRY PT ADDR -MAPFLD PRCSSNG |
| 0007A8 | GSASYAA DS | A ADDR OF SYMBOL TABLE |
| 0007AC | GSAUUIA DS | A ADDR OF UMBRELLA IDD WORK AREA |
| 0007B0 | GSAROLNM DS | CL32 ROLE NAME |
| 0007D0 | GSARRECA DS | A GRD ADDR FOR ROLE |
| 0007D4 | GSAPRECA DS | A GRD ADDR FOR PRIMARY REC FOR ROLE |
| 0007D8 | GSAROLSV DS | F REGISTER SAVE FOR BLDROL |
| 0007DC | GSAMINDX DS | H MAP RECORD INDEX FOR CURRENT REC |
| 0007DE | GSANMXDE DS | H # OF ENTRIES IN MXDE TABLE*LCB84159* |
| 0007E0 | GSAFWD DS | F FULLWORD WORK FIELD |
| 0007E4 | GSARPEP1 DS | A ADDR OF GNRP ENTRY PT |
| 0007E8 | GSANODE DS | CL8 TEMP HOLDER FOR NODENAME |
| 0007F0 | GSADICT DS | CL8 TEMP HOLDER FOR DICT DBNAME |
| 0007F8 | GSATMXDE DS | A ADDR OF MXDE TABLE *JMA85070** |
| | * | GSACSA ABOVE AND THE ADSCSA FIELD SPECIFIED BELOW BOTH |
| | * | CONTAIN THE SAME ADDRESS AND SHOULD BE INITIALIZED TOGETHER |
| 0007FC | ADSCSA DS | A #ENTER NAME FOR CSA ADDR *PHH85085* |
| 000800 | ADSIDMSA DS | A #ENTER NAME FOR IDMS ADDR *PHH85085* |
| 000804 | ADSENTP DS | 3F #ENTER NAME FOR PARMS *PHH85085* |
| 000810 | GSACMPRA DS | A ADDR OF COMPRESS BUFFER ECM86175 |
| 000814 | GSADADSA DS | A Address of ADADS block LMA |
| 000818 | GSASQCFDS DS | A ADDRESS OF SQCFRNT BLOCK *LMA87321* |
| 00081C | GSATCEA DS | A ADDRESS OF TCE *MET88081* |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|-------------------------------------|
| 000820 | DS 10F | RESERVED FOR FUTURE USE *MET88081* |
| 000848 | GSAMPRCD DS CL1200 | GENERATOR MAP RECORD AREA*JMA85070* |
| 000CF8 | *__FOLLOWING FIELDS STARTING WITH BTCH ARE BATCH GENERATOR FIELDS--* | |
| | ORG GSAMPRCD | |
| 000848 | BTCHPMAP DS CL32 | BATCH PREMAP NAME |
| 000868 | BTCHPVER DS H | BATCH PREMAP VERSION |
| 00086A | BTCHRESP DS CL32 | BATCH RESPONSE NAME |
| 00088A | BTCHRVER DS H | BATCH RESPONSE VERSION |
| 00088C | BTCHREDT DS CL3 | BATCH RESPONSE EDIT ERROR FIELD |
| 00088F | BTCHRCKY DS CL5 | BATCH RESPONSE CONTROL KEY |
| 000894 | BTCHRFNC DS CL32 | BATCH RESPONSE FUNC NAME |
| 0008B4 | BTCHINBF DS CL80 | BATCH INPUT BUFFER |
| 000904 | BTCHCOLM DS H | BATCH LEX COLUMN |
| 000906 | BTCHLNTH DS H | BATCH LEX TOKEN LENGTH |
| 000908 | BTCHADDR DS A | BATCH LEX CURRENT TOKEN ADDRESS |
| 00090C | BTCHFKEY DS X | CONVERTED BTCHRCKY |
| 00090D | BTCHNCNM DS CL32 | BATCH NEW COPY NAME |
| 00092E | BTCHNCVR DS H | BATCH NEW COPY VERSION |
| 000930 | BTCHNCNC DS CL1 | BATCH NEW COPY DENOTER |
| 000931 | BTCHNCWK DS CL1 | BATCH WORK RCD DENOTER |
| | BTCHFLG #FLAG X'80' | BATCH FLAG SET TO FAIL |
| 000932 | BTCHFLGI DS 0XL1 | |
| | 00080 BTCHFLGM EQU X'80' | |
| | BTCHNGN #FLAG X'40' | FLAG SET SO DLG CANT BE GENERATED |
| 000932 | 00040 BTCHNGNI DS 0XL1 | |
| | BTCHNGNM EQU X'40' | |
| | BTCHPRT #FLAG X'20' | FLAG SET SO LINE DONT GET PRTND 2 |
| 000932 | 00020 BTCHPRTI DS 0XL1 | |
| | BTCHPRTM EQU X'20' | |
| | BTCHNCP #FLAG X'10' | FLAG SET WHEN ADD REC NC/WK PARSED |
| 000932 | 00010 BTCHNCPI DS 0XL1 | |
| | BTCHNCPM EQU X'10' | |
| | BTCHPCP #FLAG X'08' | FLAG SET WHEN PREMAP FAILS TO COMP |
| 000932 | 00008 BTCHPCPI DS 0XL1 | |
| | BTCHPCPM EQU X'08' | |
| | BTCHERR #FLAG X'04' | FLAG SET WHEN SYNTAX ERRORS FOUND |
| 000932 | 00004 BTCHERRI DS 0XL1 | |
| | BTCHERRM EQU X'04' | |
| | GSARC8 #FLAG X'02' | SET WHEN COND CODE 8 NEEDED |
| 000932 | 00002 GSARC8I DS 0XL1 | |
| | GSARC8M EQU X'02' | |
| | GSARC4 #FLAG X'01' | SET WHEN COND CODE 4 NEEDED |
| 000932 | 00001 GSARC4I DS 0XL1 | |
| | GSARC4M EQU X'01' | |
| 000932 | BTCHFLAG DS X | |
| 000934 | DS 0H | |
| 000934 | BTCHMDIX DS X | SAVE NCBMODIX FOR INTERNAL USE |
| 000935 | BTCHRSP DS CL1 | STORES ADD/MOD/DEL OF PREMAP/RESP |
| 000936 | BTCHFRKY DS CL5 | 'FROM KEY' FOR RP MODIFY |
| 00093B | BTCHFFNC DS CL32 | 'FROM RESP VALUE' FOR RP MODIFY |
| | BTCHFRK #FLAG X'80' | IF ON, BTCHFRKY IS SET |
| 00095B | BTCHFRKI DS 0XL1 | |
| | 00080 BTCHFRKM EQU X'80' | |
| | BTCHFRF #FLAG X'40' | IF ON, BTCHFFNC IS SET |
| 00095B | 00040 BTCHFRFI DS 0XL1 | |
| | BTCHFRFM EQU X'40' | |
| | BTCHSSC #FLAG X'20' | IF ON, SUBSCHEMA CHANGED *RQE85017* |
| 00095B | 00020 BTCHSSCI DS 0XL1 | |
| | BTCHSSCM EQU X'20' | |
| | BTCHFBC #FLAG X'10' | IF ON, BTCHFRBK IS SET *DKJ85162* |
| 00095B | 00010 BTCHFBCKI DS 0XL1 | |
| | BTCHFBCKM EQU X'10' | |
| 00095B | BTCHFLG2 DS X | SECOND BATCH FLAG |
| 00095C | BTCHFRKV DS X | 'FROM KEY' INTERNAL VALUE*RQE84163* |
| 00095D | BTCHFBKY DS X | BCE INTERNAL VALUE *DKJ85162* |
| 00095E | BTCHFRBK DS X | 'FROM BCE' INTERNAL VALUE*DKJ85162* |

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|-----------------|--------------------------------------|
| 00095F | BTCHRDEF DS | CL3 | BATCH DEFAULT RESPONSE *DKJ85149* |
| 000962 | BTCHRBCE DS | CL6 | BATCH CONTROL ELEMENT *DKJ85149* |
| 000968 | BTCHFBCE DS | CL6 | 'FROM BCE' FOR RP MODIFY *DKJ85162* |
| 00096E | ORG | | |
| 004B0 | GSAMRECL EQU | *-GSAMPRCD | LENGTH OF MAP RECORD AREA |
| 000CF8 | GSAADPCH DS | F | ADDR OF ADCM'S PATCH AREA*RQE86027** |
| 000CFC | GSAGXCAD DS | F | Addr of GEXC work storage |
| 000D00 | GSAGXCSZ DS | F | Size of GEXC work storage |
| 00FA0 | GSAXCSZ EQU | 4000 | Length of ex code build area |
| 000D04 | SDRBUFA DS | A | SDR-042 BUFFER ADDRESS **SQL** |
| 000D08 | SDESBUFA DS | A | SDES-044 BUFFER ADDRESS **SQL** |
| 000D0C | DS | 7F | RESERVED FOR FUTURE USE |
| 00D28 | GSALEN EQU | ((*-GSA+3)/4)*4 | LENGTH OF GSA |
| 00D28 | GSAALEN EQU | ((*-GSA+3)/4)*4 | * LENGTH OF GSA FOR APPL GENERATOR |
| 000D28 | GSASTACK DS | 0F | START OF INTERNAL STACK |
| 000D28 | GSAASTCK DS | 0F | * START OF INT STACK FOR APPL GEN |

1.71 #GSSDS

```
COPY #GSSDS
*****
***          GSS : SUBSCHEMA AREA CONTROL BLOCK
***          ****
*****  


| <u>Offset</u> | <u>Value</u> |                   |                                   |
|---------------|--------------|-------------------|-----------------------------------|
| 000000        | GSS          | DSECT             | 11:58:42 09/30/83                 |
| 000000        | GSSNXTA      | DS A              | ADDRESS OF NEXT GSS               |
| 000004        | GSSAREAN     | DS CL32           | SUBSCHEMA AREA NAME               |
| 000024        | GSSPRLK      | DS F              | AREA PRIVACY LOCKS                |
| 000028        | GSSRDY       | DS F              | AREA READY MODE                   |
| 00002C        | GSSACTA      | DS F              | ADDR OF AREA ACTIVITY CONTROL BLK |
| 00030         | GSSLEN       | EQU *-GSS         | LENGTH OF GSS                     |
| 0000C         | GSSLENF      | EQU ((*-GSS+3)/4) | LENGTH OF GSS IN WORDS            |


```

1.72 #GTCDS

```

COPY #GTCDS
*****
***      GTC: ADS APPLICATION GENERATOR TASK CODE RECORD (INTERNAL) ***
***      GTC IS A DSECT THAT IS THE INTERNAL CONTROL BLOCK USED TO ***
***      HOLD DATA FROM EACH ADSO-APPL-TASK-CODES RECORD.
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------|
| 000000 | GTC | DSECT |
| 000000 | GTCNXTA | DS A |
| 000004 | GTCPREVA | DS A |
| 000008 | GTCSEQN | DS F |
| 00000C | GTCNAME | DS CL8 |
| 000014 | GCTOPF | DS CL8 |
| 00001C | GTCFINDX | DS H |
| 0001E | GTCLEN | EQU *-*GTC |
| 00008 | GTCLENF | EQU ((*-GTC+3)/4) |

12:17:04 04/04/85
 ADDR OF NEXT GTC
 ADDR OF PREVIOUS GTC
 TASK CODE SEQUENCE NUMBER
 TASK CODE NAME
 TOP FUNCTION FOR THIS TASK NAME
 TOP FUNCTION INDEX
 LENGTH OF GTC
 LENGTH OF GTC IN FULL WORDS

1.73 #GVCDS

```
COPY #GVCDS
*****
*** GVC IS A DSECT THAT DEFINES THE LOGICAL RECORD VERB USAGE ***
*** COUNT CONTROL BLOCK FOR THE ADS/ONLINE GENERATOR. EACH GLR ***
*** FOR A LOGICAL RECORD OWNS A LIST OF GVC'S, ONE PER PROCESS ***
*** GPR, TO MAINTAIN A COUNT OF VERB USAGE FOR THAT LOGICAL ***
*** RECORD IN EACH PROCESS. IN ADDITION, EACH GPR ANCHORS A ***
*** CHAIN THROUGH THE GVC'S FOR EACH LOGICAL RECORD REFERENCED ***
*** BY THAT PROCESS. ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------------------|
| 000000 | GVC | DSECT |
| 000000 | GVCLNXTA | DS A |
| 000004 | GVCLPRVA | DS A |
| 000008 | GVCPNXTA | DS A |
| 00000C | GVCPPRVA | DS A |
| 000010 | GVCGLRA | DS A |
| 000014 | GVCEASE | DS H |
| 000016 | GVCMODFY | DS H |
| 000018 | GVCTSTR | DS H |
| 00001A | GVCOBTN | DS H |
| 00001C | GVCCOPY | DS H |
| 00008 | GVCLENF | EQU $((*-GVC+3)/4)$ |
| 00020 | GVCLEN | EQU GVCLENF*4 |
| | | 12:43:25 12/02/81 |
| | | ADDR OF NEXT GVC FOR THE GLR |
| | | ADDR OF PRIOR GVC FOR THE GLR |
| | | ADDR OF NEXT GVC FOR THE GPR |
| | | ADDR OF PRIOR GVC FOR THE GPR |
| | | ADDR OF GLR |
| | | COUNT OF ERASE VERBS |
| | | COUNT OF MODIFY VERBS |
| | | COUNT OF STORE VERBS |
| | | COUNT OF OBTAIN VERBS |
| | | COUNT OF COPY VERBS |
| | | LENGTH IN WORDS |
| | | LENGTH IN BYTES |

1.74 #GWADS

```

COPY #GWADS
*****
*** GWA : ADS/ONLINE GENERATOR WORK AREA
*** GWA IS A DSECT THAT DEFINES THE
*** ADS GENERATOR. THIS KEPT STORAGE IS MANAGED LIKE THE
*** RECORD BUFFER BLOCKS IN THE ONLINE SYSTEM. SIZE IS GOVERNED
*** BY THE BUFFER SIZES SPECIFIED IN THE OCB. THE FIRST GWA
*** IS ALLOCATED ACCORDING TO OCBPBF SZ, ANY SECONDARY GWA'S
*** ARE THE SIZE SPECIFIED IN OCBSBFSZ. THE GWA CONTAINS ALL
*** THE CONTROL BLOCKS NEEDED TO PROCESS A DIALOG, SYSCTRL, AND
*** MRB'S FOR THE GENERATOR MAPS.
*** -----
*** A SEPARATE CHAIN OF STORAGE AREAS IS MAINTAINED FOR THE USE
*** OF THE PROCESS MODULES. THESE ARE MANAGED EXACTLY AS THE
*** GWA, BUT ARE CALLED PROCESS WORK AREAS, OR "PWA'S". THIS
*** SAME CONTROL BLOCK IS USED TO MANAGE THE PWA STORAGE
*** ROUTINE IN ADSOGNRC.
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------------|--------------------------------|
| 000000 | GWA DSECT | 08:21:53 05/28/86 |
| 000000 | GWAID DS CL4 | "GWA*" OR "PWA*" |
| 000004 | GWANXTA DS A | ADDRESS OF NEXT GWA |
| 000008 | ORG GWANXTA | |
| | GWAPRIM #FLAG X'80' | ON IF PRIMARY GWA |
| 000004 | GWAPRIMI DS 0XL1 | |
| 00080 | GWAPRIMM EQU X'80' | |
| 000004 | DS X | |
| 000005 | ORG | |
| 000008 | GWAPREVA DS A | ADDRESS OF PREVIOUS GWA |
| 00000C | GWASIZE DS F | SIZE OF THIS GWA |
| 000010 | GWAFREEA DS A | ADDRESS OF FIRST FREE SPACE |
| 00014 | GWALEN EQU ((*-GWA+3)/4)*4 | LENGTH OF GWA HEADER PORTION |
| 000000 | GWAWK DSECT | PRESENT IN PRIMARY GWA ONLY |
| 000000 | GWAGFDBA DS A | ADDRESS OF GFDB |
| 000004 | GWAFGRDA DS A | ADDRESS OF FIRST GRD |
| 000008 | GWALGRDA DS A | ADDRESS OF LAST GRD |
| 00000C | GWAGPMEA DS A | ADDRESS OF GPME |
| 000010 | GWAFGRSA DS A | ADDRESS OF FIRST GRSE |
| 000014 | GWALGRSA DS A | ADDRESS OF LAST GRSE |
| 000018 | GWACGRSA DS A | ADDRESS OF CURRENT GRS |
| 00001C | GWAFGSSA DS A | ADDRESS OF FIRST GSS |
| 000020 | GWALGSSA DS A | ADDRESS OF LAST GSS |
| 000024 | GWAFGLEA DS A | ADDRESS OF FIRST GLE IN LIT PL |
| 000028 | GWALGLEA DS A | ADDRESS OF LAST GLE IN LIT PL |
| 00002C | GWAFGLRA DS A | ADDRESS OF FIRST GLR |
| 000030 | GWALGLRA DS A | ADDRESS OF LAST GLR |
| 000034 | GWAFOFDA DS A | ADDRESS OF FIRST OFD |
| 000038 | GWALOFDA DS A | ADDRESS OF LAST OFD |
| 00003C | GWANODE DS CL8 | ALTERNATE DBNAME |
| 000044 | GWADICT DS CL8 | ALTERNATE SUBSCHEMA NAME |
| 00004C | GWAMSGS DS 4F | MESSAGE CODE TABLE |
| 00005C | GWAMSGNM DS H | CURRENT MSG TABLE POSITION |
| 00005E | GWALPOFF DS H | NEXT AVAIL. OFFSET IN LIT POOL |
| 000060 | GWANXTMD DS H | NEXT MODULE TO BE CALLED |
| 000062 | GWANXTFC DS H | FUNCTION CODE FOR NEXT MODULE |
| 000064 | GWARINDX DS H | HIGHEST EXISTING RECORD INDEX |

| | | | |
|--------|---------------------|----------|--------------------------------|
| 000066 | GWAIRAIX DS | H | RECORD INDEX FOR IRA |
| 000068 | GWALITIX DS | H | RECORD INDEX FOR LITPOOL |
| 00006A | GWAVDBIX DS | H | RECORD INDEX FOR VDB |
| 00006C | GWAVREIX DS | H | RECORD INDEX FOR VRE |
| 00006E | GWARSPNM DS | CL32 | SELECTED RESPONSE NAME |
| 00008E | ORG | GWARSPNM | |
| 00006E | GWAPGID DS | OCL26 | PARM LIST FOR #RTNCB MACRO |
| 00006E | GWAPGNM DS | CL8 | PROGRAM NAME *RZM86031043 |
| 000076 | GWAPGVER DS | H | PROGRAM VERSION *RZM86031043 |
| 000078 | GWAPNODE DS | CL8 | NODE NAME FOR PGM *RZM86031043 |
| 000080 | GWAPDICT DS | CL8 | DICT NAME FOR PGM *RZM86031043 |
| 000088 | ORG | | |
| 00008E | GWARCNDM DS | CL32 | SELECTED RECORD NAME |
| 0000AE | GWAACTN DS | X | CURRENT ACTION CODE |
| | GWADISP #FLAG X'80' | | ON IF IN DISPLAY MODE |
| 0000AF | GWADISPI DS | 0XL1 | |
| 000080 | GWADISPM EQU | X'80' | |
| | GWAFAST #FLAG X'40' | | ON IF IN FAST MODE |
| 0000AF | GWAFASTI DS | 0XL1 | |
| 000040 | GWAFASTM EQU | X'40' | |
| | GWAFSEL #FLAG X'20' | | ON IF NEXT FUNCTION SELECTED |
| 0000AF | GWAFSELI DS | 0XL1 | |
| 000020 | GWAFSELM EQU | X'20' | |
| | GWASTMP #FLAG X'10' | | ON WHEN STEP MODE MAPOUT |
| 0000AF | GWASTMPI DS | 0XL1 | |
| 000010 | GWASTMPM EQU | X'10' | |
| | GWAMSG #FLAG X'08' | | ON IF MESSAGES FOR SCREEN |
| 0000AF | GWAMSGI DS | 0XL1 | |
| 00008 | GWAMSGM EQU | X'08' | |
| | GWAMSOV #FLAG X'04' | | ON IF MORE THAN 4 MESSAGES |
| 0000AF | GWAMSOVI DS | 0XL1 | |
| 00004 | GWAMSOVM EQU | X'04' | |
| | GWACERR #FLAG X'02' | | ON WHEN COMPILE ERRORS |
| 0000AF | GWACERRI DS | 0XL1 | |
| 000002 | GWACERRM EQU | X'02' | |
| | GWANDFL #FLAG X'01' | | ON IF DIFFERENT NODE SPECIFIED |
| 0000AF | GWANDFLI DS | 0XL1 | |
| 000001 | GWANDFLM EQU | X'01' | |
| 0000AF | GWAFLAG1 DS | X | |
| | GWAFIND #FLAG X'80' | | |
| 0000B0 | GWAFINDI DS | 0XL1 | |
| 000080 | GWAFINDM EQU | X'80' | |
| | GWAGPM #FLAG X'40' | | ON IF GWACPROA POINTS TO A GPM |
| 0000B0 | GWAGPMI DS | 0XL1 | |
| 000040 | GWAGPMM EQU | X'40' | |
| | GWAPERR #FLAG X'20' | | ERROR FOUND IN LAST NC SCREEN |
| 0000B0 | GWAPERI DS | 0XL1 | |
| 000020 | GWAPERRM EQU | X'20' | |
| | GWASTSK #FLAG X'10' | | ON IF GEN2 RUNNING AS SUBTASK |
| 0000B0 | GWASTSKI DS | 0XL1 | |
| 000010 | GWASTSKM EQU | X'10' | |
| | GWALR #FLAG X'08' | | ON IF ONLY LR ACCESS ALLOWED |
| 0000B0 | GWALRI DS | 0XL1 | |
| 000008 | GWALRM EQU | X'08' | |
| | GWANOLR #FLAG X'04' | | ON IF NO LR ACCESS ALLOWED |
| 0000B0 | GWANOLRI DS | 0XL1 | |
| 000004 | GWANOLRM EQU | X'04' | |
| | GWANOPR #FLAG X'02' | | ON IF NO PRINT IN BATCH COMPLR |
| 0000B0 | GWANOPRI DS | 0XL1 | |
| 000002 | GWANOPRM EQU | X'02' | |
| | GWABTCH #FLAG X'01' | | ON IF BATCH COMPILER |
| 0000B0 | GWABTCHI DS | 0XL1 | |
| 000001 | GWABTCHM EQU | X'01' | |
| 0000B0 | GWAFLAG2 DS | X | |
| | GWAYER #FLAG X'40' | | ON IF VERSION QUALIFIERS |
| 0000B1 | GWAYERI DS | 0XL1 | |

| | | | |
|--------|-------|---------------------|---------------------------------|
| | 00040 | GWAVERM EQU X'40' | |
| | | GWAALL #FLAG X'20' | ON IF REGENNING ALL DIALOGS |
| 0000B1 | 00020 | GWAALLI DS OXL1 | |
| | | GWAALLM EQU X'20' | |
| | | GWAMAPR #FLAG X'10' | ON IF MAP RECORDS PROCESSED |
| 0000B1 | 00010 | GWAMAPRI DS OXL1 | |
| | | GWAMAPRM EQU X'10' | |
| 0000B1 | | GWABFLG1 DS X | BATCH GENERATOR FLAG BYTE |
| 0000B4 | | STMTNUM DS F | TEXT-088 SEQUENCE NUMBER |
| 0000B8 | | INBUF DS CL80 | INPUT BUFFER AREA |
| 000108 | | GWAMSGLN DS CL4 | RELATIVE LINE NUMS FOR GNNC |
| 00010C | | GWAOTBIX DS H | RECORD INDEX FOR OTB |
| 00010E | | GWAPRBCT DS H | CURRENT # OF BATCH PRINT BUFFS |
| 000110 | | GWAFPRBA DS A | ADDR OF 1ST BATCH PRINT BUFFER |
| 000114 | | GWAFDNBA DS A | ADDR OF 1ST DIALOG NAME BLOCK |
| 000118 | | GWACDNBA DS A | ADDR OF CURRENT DNB |
| 00011C | | GWAFRDBA DS A | ADDR OF 1ST REQUEST DESC BLK |
| 000120 | | GWACRDBA DS A | ADDR OF CURRENT RDB |
| 000124 | | GWAFRVBA DS A | ADDR OF 1ST REQUEST VER BLOCK |
| 000128 | | GWACRVBA DS A | ADDR OF 1ST REQUEST VER BLOCK |
| 00012C | | GWACPRBA DS A | ADDR OF CURRENT PRINT BUFFER |
| 000130 | | GWAPRDBK DS F | DBK FOR PROG-051 FOR DIALOG |
| 000134 | | GWAUERN DS CL32 | USER ID FOR USER |
| 000154 | | GWAPASSW DS CL8 | PASSWORD FOR USER |
| 00015C | | GWAFXPCA DS F | 1ST PROCESS HDR EXEC CNTL BLK |
| 000160 | | GWALPXCA DS F | LAST PROC EXECUTABLE CODE BLK |
| 000164 | | GWANFDBA DS F | ADDR OF AREA TO BUILD NEW FDB |
| 000168 | | GWANFDBS DS F | LENGTH OF NEW FDB |
| 00016C | | GWACPXCA DS F | ADDR OF CURRENT PEXC CNTL BLK |
| 000170 | | GWACOFTB DS F | CURRENT ENTRY IN OFFSET TABLE |
| | | GWATRUE #FLAG X'80' | TRUE OFFSET FND IN OFFSET TBL |
| 000174 | 00080 | GWATRUEI DS OXL1 | |
| | | GWATRUEM EQU X'80' | |
| | | GWAFLSE #FLAG X'40' | FALSE OFFSET FND IN OFFSET TBL |
| 000174 | 00040 | GWAFLSEI DS OXL1 | |
| | | GWAFLSEM EQU X'40' | |
| | | GWAPRGF #FLAG X'20' | DIALOG FAILED PROG REGIST. |
| 000174 | 00020 | GWAPRGFI DS OXL1 | |
| | | GWAPRGFM EQU X'20' | |
| | | GWAASEC #FLAG X'10' | ADS PRODUCT SECURITY IS ON |
| 000174 | 00010 | GWAASECI DS OXL1 | |
| | | GWAASECM EQU X'10' | |
| 000174 | | GWACMTST DS X | FLAG BYTE |
| 000175 | | GWAUNXF DS X | NEXT TASK FLAG (UMBRELLA) |
| 000176 | | GWAMSWIX DS H | REC INDEX FOR MAP SWITCH REC |
| 000178 | | GWAUCEA DS A | ADDR OF SUBTASK PARAMETER LIST |
| 00017C | | GWAUDBK DS A | DBKEY OF USER-047 |
| 000180 | | GWANFDBF DS F | NEW FDB BUFFER SIZE *LCB84117* |
| 000184 | | GWAUNXT DS CL8 | NEXT TASK CODE (UMBRELLA) |
| 00018C | | GWARCDVR DS H | SELECT RECORD VERSION (UMBR) |
| 00018E | | GWAPUBL DS X | PUBL051 VALUE IN PROG-051 |
| | | GWAOLM #FLAG X'80' | ONLINE MAP (GNDB 15) *JMA85078* |
| 00018F | 00080 | GWAOLMI DS OXL1 | |
| | | GWAOLMM EQU X'80' | |
| | | GWAINPM #FLAG X'40' | INPUT MAP (GNDB 15) *JMA85078* |
| 00018F | 00040 | GWAINPMI DS OXL1 | |
| | | GWAINPMM EQU X'40' | |
| | | GWAOUTM #FLAG X'20' | OUTPUT MAP (GNDB 15) *JMA85078* |
| 00018F | 00020 | GWAOUTMI DS OXL1 | |
| | | GWAOUTMM EQU X'20' | |
| | | GWAPRSI #FLAG X'08' | INPUT PARSED |
| 00018F | 00008 | GWAPRSII DS OXL1 | |
| | | GWAPRSIM EQU X'08' | |
| | | GWAPRSO #FLAG X'04' | OUTPUT PARSED |
| 00018F | 00004 | GWAPRSOI DS OXL1 | |
| | | GWAPRSOM EQU X'04' | |

| | | |
|--------|--------------------------|----------------------------------|
| | GWAPRSS #FLAG X'02' | SUSFILE PARSED |
| 00018F | GWAPRSSI DS 0XL1 | |
| | 00002 GWAPRSM EQU X'02' | |
| 00018F | GWAFLAG3 DS X | FLAG BYTE *JMA85078* |
| 000190 | GWASTKST DS A | MESSAGE PARM STACK START ADDR |
| 000194 | GWASTKCT DS F | MESSAGE PARM STACK COUNT |
| | GWAXIRC #FLAG X'80' | AN EXTERNAL REC FOR AN*JMA85344* |
| 000198 | GWAXIRCI DS 0XL1 | |
| | 00080 GWAXIRCM EQU X'80' | |
| | * | OUTPUT MAP IS ALSO AN *JMA85344* |
| | * | INTERNAL RECORD *JMA85344* |
| | GWAXREC #FLAG X'40' | AN EXTERNAL REC FOR AN*JMA85344* |
| 000198 | GWAXRECI DS 0XL1 | |
| | 00040 GWAXRECM EQU X'40' | |
| | * | OUTPUT MAP IS NOT USED*JMA85344* |
| | * | AS AN INTERNAL RECORD *JMA85344* |
| | GWACOPY #FLAG X'20' | COPY FROM SESSION *MET87215* |
| 000198 | GWACOPYI DS 0XL1 | |
| | 00020 GWACOPYM EQU X'20' | |
| | GWAWNDO #FLAG X'10' | |
| 000198 | GWAWNDOI DS 0XL1 | GNRP IN WINDOW MODE *MET87219* |
| | 00010 GWAWNDOM EQU X'10' | |
| | GWAMAP2 #FLAG X'08' | ALTERNATE WINDOW MAP *MET87243* |
| 000198 | GWAMAP2I DS 0XL1 | |
| | 00008 GWAMAP2M EQU X'08' | |
| | GWAPSRC #FLAG X'04' | PSRC SCRATCH EXISTS *MET87342* |
| 000198 | GWAPSRCI DS 0XL1 | |
| | 00004 GWAPSRCM EQU X'04' | |
| | GWQUOT #FLAG X'02' | WITHIN QUOTED STRING *ECM87272* |
| 000198 | GWQUOTI DS 0XL1 | |
| | 00002 GWQUOTM EQU X'02' | |
| | GWARSUM #FLAG X'01' | RESUME AFTER OTP CALL *MET87295* |
| 000198 | GWARSUMI DS 0XL1 | |
| | 00001 GWARSUMM EQU X'01' | |
| 000198 | GWAFLAG4 DS X | FLAG BYTE *JMA85344* |
| | GWAGERR #FLAG X'80' | GENERATE ERROR *MET88076* |
| 000199 | GWAGERRI DS 0XL1 | |
| | 00080 GWAGERRM EQU X'80' | |
| | GWAPEND #FLAG X'40' | DROP PENDING *MET88076* |
| 000199 | GWAPENDI DS 0XL1 | |
| | 00040 GWAPENDM EQU X'40' | |
| | GWADLM #FLAG X'20' | COMMENT STATEMENT GIK91081032 |
| 000199 | GWADLMI DS 0XL1 | |
| | 00020 GWADLMM EQU X'20' | |
| 000199 | GWAFLAG5 DS X | FLAG BYTE *MET88076* |
| 00019A | GWAPROD DS X | PRODUCT VERSION *MET88076* |
| | 00001 GWAIDMS EQU 1 | .. IDMS *MET88076* |
| | 00002 GWAWSAM EQU 2 | .. VSAM *MET88076* |
| | 00003 GWADB2 EQU 3 | .. DB2 *MET88076* |
| 00019B | GWAMPDTYP DS X | MODULE (AND SRE) TYPE *MET88067* |
| | 00006 GWADCTYP EQU 6 | .. DECLARATION *MET88067* |
| | 00002 GWAPMTYP EQU 2 | .. PREMAP *MET88067* |
| | 00003 GWAIRSTYP EQU 3 | .. RESPONSE *MET88067* |
| 00019C | GWAXGRDA DS F | ADDR OF GRD FOR EXTERN*JMA85344* |
| | * | REC FOR OUTPUT MAP *JMA85344* |
| | * | (FOR GNRC & GFDB TO *JMA85344* |
| | * | SET UP INIT COMPRESSED*JMA85344* |
| | * | RECORD) *JMA85344* |
| 0001A0 | GWASIL DS H | LENGTH OF SHIFTIN BZJ |
| 0001A2 | GWASOL DS H | LENGTH OF SHIFTOUT BZJ |
| 0001A4 | GWASO DS XL3 | SHIFTOUT SEQUENCE BZJ |
| 0001A7 | GWASI DS XL3 | SHIFTIN SEQUENCE BZJ |
| | GWADBCS #FLAG X'40' | ON IF IN DBCS FIELD BZJ |
| 0001AA | GWADBCSI DS 0XL1 | |
| | 00040 GWADBCSM EQU X'40' | |

| | | | |
|--------|----------|--|---|
| 0001AA | GWAGLT | #FLAG X'80' | ON IF PROCESING G-LITERAL BZJ |
| | GWAGLTI | DS 0XL1 | |
| 00080 | GWAGLTM | EQU X'80' | |
| | GWADBCA | #FLAG X'20' | ON IF DBCS SUPPORT ACTIVE BZJ |
| 0001AA | GWADBCAI | DS 0XL1 | |
| 00020 | GWADBCAM | EQU X'20' | |
| | GWAGLER | #FLAG X'10' | G-LITERAL ERROR FLAG *TBL89009* |
| 0001AA | GWAGLERI | DS 0XL1 | |
| 00010 | GWAGLERM | EQU X'10' | |
| | GWASRRC | #FLAG X'08' | On -> Record spec'd for Send/Receive *MCM88071* |
| 0001AA | GWASRRCI | DS 0XL1 | |
| 00008 | GWASRRCM | EQU X'08' | |
| | * | Off -> Element spec'd for Send/Receive*MCM88071* | |
| 0001AA | GWADBCSP | DS X | FLAG BYTE FOR DBCS SUPPORT BZJ |
| 0001AB | GWAMLINE | DS X | MAP LINE INDEX *MET87244* |
| 0001AC | GWAWIRA | DS A | TOOL BOX WIR RCD ADDR *MET87202* |
| 0001B0 | GWASCAIX | DS H | SQL COMM AREA INDEX *LMA88030* |
| 0001B2 | GWASCALN | DS H | AND LENGTH *LMA88040* |
| 0001B4 | GWAPBFIK | DS H | SQL PBUFF INDEX *LMA88030* |
| 0001B6 | GWAPBFNL | DS H | AND LENGTH *LMA88040* |
| 0001B8 | GWADBFIX | DS H | SQL DBUFF INDEX *LMA88030* |
| 0001BA | GWADBFNL | DS H | AND LENGTH *LMA88040* |
| 0001BC | GWAKEID | DS OCL16 | QUEUE ID FOR SUSPEND *MET88064* |
| 0001BC | GWAQDNAM | DS CL8 | DIALOG NAME *MET88064* |
| 0001C4 | GWAQDVER | DS CL4 | DIALOG VERSION *MET88064* |
| 0001C8 | GWAQTOOL | DS CL4 | TOOL ID -- \$DLG *MET88064* |
| 0001CC | GWAGDCLA | DS F | ADDRESS OF GDCL *MET88064* |
| 0001D0 | GWAPSRID | DS CL8 | PRINT SOURCE SCR ID *MET88064* |
| 0001D8 | GWAFFD@ | DS A | Address of first GRD/GRE on chain *MCM88064* |
| | * | to be used to generate FDEs in *MCM88064* | |
| | * | the FDB. *MCM88064* | |
| 0001DC | GWALFD@ | DS A | Address of last GRD/GRE on chain *MCM88064* |
| | * | to be used to generate FDEs in *MCM88064* | |
| | * | the FDB. *MCM88064* | |
| 0001E0 | GWANFDO | DS F | Next offset available in FDE area *MCM88064* |
| | * | of FDB. *MCM88064* | |
| 0001E4 | GWASGRDA | DS F | Address of GRD for send/receive arg *MCM88081* |
| 0001E8 | GWASGRE4 | DS F | Address of GRE for send/receive arg *MCM88081* |
| 0001EC | GWACHRL | DS H | LENGTH OF A DBCS CHAR *MET89025* |
| 0001EE | | DS H | RESERVED *MET89025* |
| 0001F0 | GWAPLIDA | DS A | ADDRESS OF 80 BYTE STG PRIMED WITH *JMA91143* |
| | * | PROCESS INFO FOR PRINT BUF MAINTNANCE*JMA91143* | |
| 0001F4 | | DS 5F | RESERVED *JMA91143* |
| 00208 | GWAWKLEN | EQU ((*-GWAWK+3)/4)*4 | LENGTH OF GWA WORK DSECT |
| 00082 | GWAWKLNF | EQU ((*-GWAWK+3)/4) | LENGTH OF GWA WORK IN WORDS |

1.75 #HCHDS

```

COPY #HCHDS
*****
**      HELP CONTROL HEADER      **
*****
*   Defines the existence of a Help module.          *
*   A Help module is named by the map during map compilation.  *
*   There is one Help module per map.                 *
*
*****
Offset Value
000000    HCH      DSECT           11/13/89 10:02:10  05/15/91
           *
000000    HCHNAME  DS   CL8      HELP MODULE NAME
000008    HCHVERID DS  0CL16    COMPILE DATE AND TIME
000008    HCHDATE   DS  CL8     COMPILE DATE
000010    HCHTIME   DS  CL6     COMPILE TIME
000016    HCHID     DS  CL2     ID (RELEASE)
000018    HCHLEN    DS  F       LENGTH OF HELP LOAD MODULE
00001C    HCHNFLDS DS  H       NUMBER OF HFCBS
           *
           *
00001E    HCHFLAG   DS  X       HELP FLAG BYTE 1
00001F           DS  CL21    AVAILABLE
00034     HCHBSLEN EQU  *-*HCH  FIXED LENGTH OF HCH
000000    HCE      DSECT
           *
           *****
**      HELP CONTROL ELEMENT      **
*****
*   REPRESENTS HELP TEXT FOR ONE FIELD OR THE MAP ITSELF.      *
*   THE ROW/COLUMN HERE WILL MATCH THE MCE IT HELPS OR IT      *
*   WILL CONTAIN LOW VALUES IF REPRESENTING THE MAP.           *
*
*****
000000    HCELEN   DS  H       LENGTH OF ELEMENT
           HCEMAPL  #FLAG X'80'    MAP (SCREEN) LEVEL
000002    00080    HCEMAPLI DS  0XL1
           HCEMAPLM EQU  X'80'
           HCEDFLD  #FLAG X'40'    FIELD LEVEL
000002    00040    HCEDFLDI DS  0XL1
           HCEDFLDM EQU  X'40'
           HCEFULL  #FLAG X'20'    FULL SCREEN
000002    00020    HCEFULLI DS  0XL1
           HCEFULLM EQU  X'20'
           HCELAST  #FLAG X'10'    LAST HCE IN THE HCH
000002    00010    HCELASTI DS  0XL1
           HCELASTM EQU  X'10'
           HCEFLAG   DS  X       GENERAL HCE FLAG
           *
           *
000003    00080    HCEREC   #FLAG X'80'    RECORD COMMENTS
           HCERECI  DS  0XL1
           HCERECM  EQU  X'80'
           HCEMAP   #FLAG X'40'    MAP COMMENTS
000003    00040    HCEMAPI DS  0XL1
           HCEMAPM  EQU  X'40'
           HCEREE   #FLAG X'20'    RECORD-ELEMENT COMMENTS
000003    00020    HCEREEI  DS  0XL1
           HCEREMM  EQU  X'20'
           HCEEC    #FLAG X'10'    ELEMENT COMMENTS
000003    00003    HCEECI   DS  0XL1

```

| | | | | | |
|--------|-------|----------|-------|-------|-----------------------------------|
| | 00010 | HCEECM | EQU | X'10' | |
| | | HCEMI | #FLAG | X'08' | DDLDCMSG OBTAIN |
| 000003 | | HCEMII | DS | 0XL1 | |
| | 00008 | HCEMIM | EQU | X'08' | |
| | | HCEETV | #FLAG | X'04' | EDIT TABLE LOAD |
| 000003 | | HCEETVI | DS | 0XL1 | |
| | 00004 | HCEETVM | EQU | X'04' | |
| | | HCECTV | #FLAG | X'02' | CODE TABLE LOAD |
| 000003 | | HCECTVI | DS | 0XL1 | |
| | 00002 | HCECTVM | EQU | X'02' | |
| | | HCEAT | #FLAG | X'01' | TEXT ENTERED WITH OLM |
| 000003 | | HCEATI | DS | 0XL1 | |
| | 00001 | HCEATM | EQU | X'01' | |
| 000003 | | HCEORG1 | DS | X | ORIGIN OF HELP DISPLAYED |
| | | * | | | |
| | | HCEINC | #FLAG | X'80' | INCLUDED IDD MODULE TEXT |
| 000004 | | HCEINCI | DS | 0XL1 | |
| | 00080 | HCEINCM | EQU | X'80' | |
| | | HCESUBS | #FLAG | X'40' | SUBORDINATE SUBSCRIPTED HCE |
| 000004 | | HCESUBSI | DS | 0XL1 | |
| | 00040 | HCESUBSM | EQU | X'40' | |
| 000004 | | HCEORG2 | DS | X | ORIGIN OF HELP DISPLAYED |
| 000005 | | | DS | C | AVAILABLE |
| 000006 | | HCESUBRC | DS | XL2 | SUBORDINATE SUBSCRIPT R/C POINTER |
| 000008 | | HCEROW | DS | X | ROW |
| 000009 | | HCECOL | DS | X | COLUMN |
| 00000A | | HCETEXTL | DS | H | LENGTH OF TEXT |
| 00000C | | HCEMOD | DS | CL32 | IDD MODULE NAME |
| 00002C | | HCEMODV | DS | H | IDD MODULE VERSION |
| 00002E | | | DS | CL10 | AVAILABLE |
| 000038 | | HCETEXT | DS | 0C | HELP TEXT AND COMMENTS BEGINS * |
| 00038 | | HCEBSLN | EQU | *-HCE | LENGTH OF HCE BASE |

1.76 #HVTDS

```
COPY #HVTDS
*****
***      HVTNTRY - HOST VARIABLE TABLE ENTRY (SQL)      ***
***      THE FIRST 12 BYTES OF THIS DSECT CORRESPOND TO   ***
***      AN ENTRY IN THE HOST VARIABLE TABLE THAT IS PASSED ***
***      TO THE DATABASE FOR PROCESSING. THE REMAINING PORTION ***
***      OF THE DSECT ARE FIELDS THAT ARE SAVED ALONG WITH THE ***
***      HOST VARIABLE INFORMATION WHEN THE DATABASE TELLS US   ***
***      TO DEFER THE BUILDING OF THE SQL CALL.           ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------------------|
| 000000 | HVTNTRY | DSECT |
| 000000 | HVTLEN | DS F |
| 000004 | HVTDTYP | DS H |
| 000006 | HVTSCA | DS H |
| 000008 | HVTPRE | DS H |
| 00000A | HVTALN | DS H |
| 00000C | HVTIGREA | DS A |
| 000010 | HVTIGRD | DS CL32 |
| 000030 | HVTIGRE | DS CL32 |
| 000050 | HVTOGRE | DS CL32 |
| 000070 | HVTOIND | DS H |
| 00074 | HVTLNGTH | EQU (((*-HVTNTRY+3)/4)*4) |
| | | 09/10/90 |
| | | LENGTH OF HOST VAR DESCRIPTOR |
| | | DATATYPE OF HVD |
| | | SCALE OF HVD |
| | | PRECISION OF HVD |
| | | ALIGNMENT (ALWAYS 0) |
| | | ADDRESS OF INPUT GRE |
| | | HOST VAR INPUT GRDNAME |
| | | HOST VAR INPUT GRENAME |
| | | OUTPUT GRENAME |
| | | OUTPUT GRDindx |

1.77 #INCDS

```

COPY #INCDS
*****
***      INCLUDES: INCLUDE STATEMENT SAVE AREA
***      INC IS A DSECT THAT DESCRIBES THE TEMPORARY WORK AREA
***      NEEDED FOR SAVING THE PROCESSING ENVIRONMENT WHEN AN
***      INCLUDE STATEMENT IN ENCOUNTER AS A COMMAND IN A PROCESS.
***      EACH TIME AN INCLUDE STATEMENT IS ENCOUNTERED, THE
***      FOLLOWING INFORMATION MUST BE SAVED BEFORE CONTINUING:
***      1 - RESPONSE PROCESS NAME & VERSION ON INCLUDE CARD
***          (TO PREVENT LOOPING BY INCLUDING A RESPONSE
***          PROCESS WITHIN ITSELF)
***      2 - DATABASE KEY OF THE CURRENT TEXT-088 RECORD
***      3 - FLAG TO TELL WHETHER OR NOT PROCESS IN GMD CHAIN
***      4 - DATE CREATED/LAST UPDATED FOR PROCESS
*****
*****
```

Offset Value

| | | | |
|--------|----------|----------------|-------------------------------|
| 000000 | INCENTRY | DSECT | 13:57:12 07/27/84 |
| 000000 | INCDBK | DS F | DATABASE KEY OF TEXT-088 REC |
| 000004 | INCMBBK | DS F | DB KEY OF INCLUDED MODULE-067 |
| 000008 | INCPRNAM | DS CL32 | PROCESS NAME FOR THIS INCLUDE |
| 000028 | INCPRVER | DS H | PROCESS VERSION FOR INCLUDE |
| | INCGMD | #FLAG X'80' | PROCESS ALREADY IN GMD CHAIN |
| 00002A | INCGMDI | DS OXL1 | |
| 000080 | INCGMDM | EQU X'80' | |
| 00002A | INCFLAG1 | DS X | GENERAL PURPOSE FLAG |
| 00002B | | DS X | UNUSED |
| 00002C | INCDATCR | DS CL8 | DATE CREATED |
| 000034 | INCDATLU | DS CL8 | DATE LAST UPDATED |
| 0003C | INCENTLN | EQU *-INCENTRY | LENGTH OF ONE SAVED ENTRY |

1.78 #LGSDS

```
#LGSDS
*****
*** LGS -- WORKING STORAGE FOR THE BATCH LOG SYSTEM LOAD MODULE
*** ADSOBLOG. CURRENTLY, THE PROGRAMS THAT USE IT ARE
*** ADSOLOGR AND ADSOLOS (ADSOLOS UNDER DOS).
*** THIS IS "ID" STORAGE WHICH IS AVAILABLE
*** TO THE LOGGER DURING THE ENTIRE BATCH RUN.
***
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--------------------------------------|
| 000000 | LGS | DSECT |
| 000000 | LGSID | DS CL4'LGS*' 04/30/90 12:33:21 |
| 000004 | SAVEREGS | DS 5F EYECATCH |
| 000018 | SYSPLIST | DS 12F SAVE REGISTERS |
| 000048 | | DS F PLIST |
| 00004C | LGSPRADR | DS F RESERVED |
| 000050 | LGSLoba | DS F ADDR OF LOG REC PASSED TO US |
| 000054 | LGSCCECB | DS 3F ADDR OF LOG OPTIONS BLOCK |
| 000060 | LGSIIECB | DS 3F COMPLN ECB FOR ARCHIVE SUBTASK |
| 00006C | LGSWECB | DS 3F IDLE ECB FOR ARCHIVE SUBTASK |
| 000078 | LGSTCBA | DS F WAIT ECB FOR LOGR TASK |
| 00007C | LGSRCDCT | DS F TASK CONTROL BLK FOR ARCHIVER |
| 000080 | LGSI1THR | DS F LOG RECORD COUNT |
| 000084 | LGSI1DDNM | DS CL8 LOG 1 RECORD THRESHOLD |
| 00008C | LGSI1BRPA | DS F LOG 1 DD NAME |
| 000090 | LGSI2THR | DS F LOG 1 BRPL ADDRESS |
| 000094 | LGSI2DDNM | DS CL8 LOG 2 RECORD THRESHOLD |
| 00009C | LGSI2BRPA | DS F LOG 2 DD NAME |
| 0000A0 | LGSCTHR | DS F LOG 2 BRPL ADDRESS |
| 0000A4 | LGSCDDNM | DS CL8 CURRENT LOG RECORD THRESHOLD |
| 0000AC | LGSCBRPA | DS F CURRENT LOG DD NAME |
| 0000B0 | LGSSVR7 | DS F CURRENT LOG BRPL ADDRESS |
| 0000B4 | LGSSV2R7 | DS F SAVE REGISTER 7 |
| 0000B8 | LGSBOS7 | DS F SAVE REGISTER 7 |
| 0000BC | | DS F SAVE REGISTER 7 (BOPN ROUTINE) |
| 0000C4 | LGSPRLEN | DS H RESERVED |
| 0000C6 | LGSLLEN | DS H RECORD LEN (MAY BE COMPRESSED) |
| 0000C8 | LGSBRPLN | DS H LENGTH OF RECORD TO #BPUT |
| 0000CA | LGSPPRNA | DS CL8 LENGTH OF STG TO GET FOR BRPL |
| 0000D2 | LGSPUSID | DS CL32 PROGRAM NAME FOR LOG PREFIX |
| 0000F2 | LGSPRTYP | DS XL1 USER ID FOR LOG PREFIX |
| 0000F3 | LGSCC | DS XL1 RECORD TYPE FOR LOG PREFIX |
| 0000F4 | LGSVARI | DS 0XL1 CARRIAGE CONTROL |
| 0000F4 | LGSVARM | EQU X'80' |
| 0000F4 | LGSCMPRI | DS 0XL1 |
| 0000F4 | LGSCMPRM | EQU X'40' |
| 0000F4 | LGSPREFI | DS 0XL1 |
| 0000F4 | LGSPREFM | EQU X'20' |
| 0000F4 | LG SARCI | DS 0XL1 |
| 0000F4 | LG SARCM | EQU X'10' |
| 0000F4 | LG SALTI | DS 0XL1 |
| 0000F4 | LG SALTM | EQU X'08' |
| 0000F4 | LG SI1USEI | DS 0XL1 |
| 0000F4 | LG SI1USEM | EQU X'04' |
| 0000F4 | LG SNONEI | DS 0XL1 |
| 0000F4 | LG SNONEM | EQU X'02' |
| 0000F4 | LG SCMPYI | DS 0XL1 |

| | | | | |
|--------|--------|--------------|-------|-----------------------------------|
| 0000F4 | 00001 | LGSCMPYM EQU | X'01' | |
| 0000F5 | | LGSFLG1 DS | XL1 | FLAG BYTE 1 |
| 0000F5 | 00080 | LGSARIPI DS | 0XL1 | |
| 0000F5 | | LGSINITI DS | 0XL1 | |
| 0000F5 | 00040 | LGSINITM EQU | X'40' | |
| 0000F5 | | LGSERRI DS | 0XL1 | |
| 0000F5 | 00020 | LGSERRM EQU | X'20' | |
| 0000F5 | | LGSERRI DS | 0XL1 | |
| 0000F5 | 00010 | LGSERRM EQU | X'10' | |
| 0000F5 | | LGSPRNTI DS | 0XL1 | |
| 0000F5 | 00008 | LGSPRNTM EQU | X'08' | |
| 0000F5 | | LGSBPRMI DS | 0XL1 | |
| 0000F5 | 00004 | LGSBPRMM EQU | X'04' | |
| 0000F5 | | LGSDTMSI DS | 0XL1 | |
| 0000F5 | 00002 | LGSDTMSM EQU | X'02' | |
| 0000F5 | | LGSAFNFI DS | 0XL1 | |
| 0000F5 | 00001 | LGSAFNFM EQU | X'01' | |
| 0000F5 | | LGSFLG2 DS | XL1 | FLAG BYTE 2 |
| 0000F6 | | LGSINPLI DS | 0XL1 | |
| 0000F6 | 00080 | LGSINPLM EQU | X'80' | |
| 0000F6 | | LGSNOSTI DS | 0XL1 | |
| 0000F6 | 00040 | LGSNOSTM EQU | X'40' | |
| 0000F6 | | LGSBRERI DS | 0XL1 | |
| 0000F6 | 00020 | LGSBRERM EQU | X'20' | |
| 0000F6 | | LGSIOERI DS | 0XL1 | |
| 0000F6 | 00010 | LGSIOERM EQU | X'10' | |
| 0000F6 | | LGSLGERI DS | 0XL1 | |
| 0000F6 | 00008 | LGSLGERM EQU | X'08' | |
| 0000F6 | | LGSARCHI DS | 0XL1 | |
| 0000F6 | 00004 | LGSARCHM EQU | X'04' | |
| 0000F6 | | LGSNTSKI DS | 0XL1 | |
| 0000F6 | 00002 | LGSNTSKM EQU | X'02' | |
| 0000F6 | | LGSFLG3 DS | XL1 | FLAG BYTE 3 |
| 0000F7 | | LGSNOARI DS | 0XL1 | |
| 0000F7 | 00080 | LGSNOARM EQU | X'80' | |
| 0000F7 | | LGSSTOPI DS | 0XL1 | |
| 0000F7 | 00040 | LGSSTOPM EQU | X'40' | |
| 0000F7 | | LGSFLG4 DS | XL1 | FLAG BYTE 4 *RQE86108* |
| 0000F8 | | DS | OD | |
| 0000F8 | | LGSBUFF DS | CL316 | LOG BUFFER |
| 000234 | | LGSCMPWK DS | XL256 | WORK AREA FOR COMPRESS |
| 000334 | | LGSABRPL DS | A | ADDR OF BRPL FOR ARCHIVER |
| 000338 | | LGSABUFR DS | A | ADDR OF BUFFER FOR ARCHIVER |
| 00033C | | LGSABLEN DS | H | LENGTH OF BRPL FOR ARCHIVER |
| 00033E | | LGSADDNM DS | 0CL8 | DDNAME OF CURRENT ARCHIVE FILE |
| 00033E | | LGSADDAL DS | CL6 | 'ARCLOG' |
| 000344 | | LGSADDNU DS | CL1 | <NUMERIC> IN DISPLAY (1-9) |
| 000345 | | LGSADDSP DS | CL1 | SPACE |
| 000346 | | LGSADDNL DS | CL1 | ARCLOG FILE LIMIT(1-9)*RQE86107* |
| 000347 | | LGSRCL DS | X | LENGTH OF PARM FIELD |
| 000348 | | LGSRC DS | CL2 | RETURN CODE PARM |
| 000350 | | LGSDBWD DS | D | DOUBLE WORD FOR WORK |
| 000358 | | LGSSBRP1 DS | F | COPY OF LOG1 BRPL *DKJ85343* |
| 00035C | | LGSSBRP2 DS | F | COPY OF LOG2 BRPL *DKJ85343* |
| 000360 | | LGSPTCHA DS | A | PATCH AREA ADDRESS |
| 000364 | | LGSH DS | F | SAVE AREA FOR BIN HOURS*LCB85297* |
| 000368 | | LGSM DS | F | SAVE AREA FOR BIN MINUT*LCB85297* |
| 00036C | | LGSS DS | F | SAVE AREA FOR BIN SECND*LCB85297* |
| 000370 | | LGSFWD DS | F | FULLWORD FOR WORK |
| 000374 | | DS | 2F | RESERVED |
| 0037C | LGSLEN | EQU | *-LGS | LENGTH OF LGS DSECT |

1.79 #LMTDS

```
COPY #LMTDS
*****
***          LMT: LOAD MODULE TABLE ENTRY
***          LMT IS A DSECT THAT DESCRIBES AN ENTRY IN THE LOAD MODULE
***          TABLE ASSOCIATED WITH USER FUNCTIONS. THERE IS ONE ENTRY
***          CREATED FOR EACH XDE MODEL TABLE LOADED DURING THE PROCESSING
***          OF BUILT-IN AND USER FUNCTIONS.
***          ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-----------|
| 000000 | LMT | DSECT |
| 000000 | LMTNAME | DS CL8 |
| 000008 | LMTADDR | DS A |
| 00000C | LMTLEN | EQU *-LMT |

17:46:26 05/30/84
NAME OF XDE MODEL TABLE LOAD MODUL
ADDR OF XDE MODEL TABLE " "
LENGTH OF 1 ENTRY IN LOAD MOD TBL

1.80 #LNTDS

```
COPY #LNTDS
*****
***          LNT: LINE NUMBER TABLE ENTRY
***          ***
***  LNT IS A DSECT THAT DESCRIBES AN ENTRY IN THE LINE NUMBER  ***
***  TABLE LOCATED IN THE FDB.  THERE IS ONE ENTRY FOR EACH CME.  ***
***          ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------|
| 000000 | LNT | DSECT |
| 000000 | LNTSEQ | DS XL4 |
| 000004 | LNTMODIX | DS XL2 |
| 00006 | LNTLEN | EQU *-LNT |

11:35:26 02/23/83
IDB SOURCE LINE NUMBER
INDEX INTO INCLUDED MODULE TABLE
LENGTH OF 1 ENTRY IN LINE # TABLE

1.81 #LOBDS

```
COPY #LOBDS
*****
*** LOB: LOG OPTIONS BLOCK FOR BATCH
***
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------|------------------------------------|
| 000000 | LOB DSECT | 16:03:15 02/05/86 |
| 000000 | LOBID DS CL4 | 'LOB*' *** |
| 000004 | LOBF1CT DS F | FILE 1 COUNT *** |
| 000008 | LOBF2CT DS F | FILE 2 COUNT *** |
| 00000C | DS F | RESERVED *DKJ86036* |
| 000010 | DS H | RESERVED *DKJ86036* |
| 000012 | LOBF1NAM DS CL8 | FILE 1 NAME |
| 00001A | LOBF2NAM DS CL8 | FILE 2 NAME |
| 000022 | LOBFLG1 DS X | FIRST FLAG BYTE |
| | LOBARCH #FLAG X'80' | IF ON, ARCHIVING DESIRED (OS ONLY) |
| 000023 | LOBARCHI DS 0XL1 | |
| 000023 | 00080 LOBARCHM EQU X'80' | COMPRESS LOG RECORDS |
| | LOBCMPR #FLAG X'40' | |
| 000023 | 00040 LOBCMPRI DS 0XL1 | |
| | 00040 LOBCMPRM EQU X'40' | |
| | LOBNOCM #FLAG X'20' | DO NOT COMPRESS LOG RECORDS |
| 000023 | 00020 LOBNOCMI DS 0XL1 | |
| | 00020 LOBNOCMM EQU X'20' | |
| | LOBPREF #FLAG X'10' | WRITE LOG RECORD PREFIX |
| 000023 | 00023 LOBPREFI DS 0XL1 | |
| | 00010 LOBPREFM EQU X'10' | |
| | LOBNOPR #FLAG X'08' | DO NOT WRITE LOG RECORD PREFIX |
| 000023 | 00008 LOBNOPRI DS 0XL1 | |
| | 00008 LOBNOPRM EQU X'08' | |
| 000023 | 000023 LOBFLG2 DS X | SECOND FLAG BYTE |
| 000024 | DS 4F | RESERVED |
| 00034 | LOBLEN EQU *-LOB | LENGTH OF LOG OPTIONS BLOCK |

1.82 #MAPTDS

```

COPY #MAPTDS
*****
***      #MAPTDS - MAP-TYPE TABLE DSECT      ***
***      USED FOR BUILDING AND REFERENCING MAP-TYPES   ***
***      THAT WILL ASSOCIATE SIMILAR MAPS TO EACH OTHER. ***
*****
Offset  Value
000000  MAPTAB    DSECT          14:44:02 04/17/84
000000  MAPTNXT   DS   A          DISPLACEMENT TO NEXT MAP-TYPE TABLE
000000  *           DS   H          F'0' = LAST TABLE ENTRY
000004  MAPTCNT   DS   H          NUMBER OF FROM/TO ENTRIES IN * TABLE
000006  MAPTFL1   DS   X          FLAG BYTE 1
000007  DS         DS   X          RESERVED
000008  MAPTTYPE  DS   CL8         MAP-TYPE NAME
000010  MAPTNAM1  DS   CL8         "FROM" MAP-NAME
000018  MAPTNAM2  DS   CL8         "TO"   MAP-NAME
00010  MAPTNAML EQU  *-MAPTNAM1 LENGTH FOR MULTIPLE ENTRIES
00020  MAPTABL   EQU  *-MAPTAB

```

1.83 #MASKDS

```
COPY #MASKDS
*****
*** MASKDS IS THE DSECT USED FOR COBOL MOVE COMPATIBILITY WITH ***
*** PICTURE TO EDIT CONVERSION ***
*****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-------------------|--------------------------------|
| 000000 | MASK DSECT | 12:14:35 05/15/84 |
| 000000 | MASKXDE DS 0CL6 | |
| 000000 | MASKPWR DS C | POWER 10-1 |
| 000001 | MASKSGN DS C | LEADING SIGN CHAR (\$,-,+,...) |
| 000002 | MASKPLN DS 0C | PICTURE LEN |
| 000002 | MASKNDEC DS C | # OF DECIMALS |
| 000003 | MASKDATY DS X | DATA TYPE |
| 000004 | MASKPADN DS C | PAD/FILL CHARACTER |
| 000005 | MASKFLG DS X | EDIT FLAGS |
| 000006 | MASKLEN DS H | LENGTH OF MASK |
| 000008 | MASKMASK DS CL256 | MASK (IF CREATED) |

1.84 #MDTDS

```
COPY #MDTDS
*****
***          MDT: INCLUDE MODULE TABLE ELEMENT
***          MDT IS A DSECT THAT DESCRIBES A PROCESS INCLUDED VIA
***          AN 'INCLUDE' COMMAND IN ANOTHER PREMAP OR RESPONSE PROCESS.
***          THIS INFORMATION IS USED TO KEEP TRACK OF DEBUGGING
***          INFORMATION IN THE FDB FOR THE NEW RUN-TIME DEBUGGING SCREEN.
***          ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-----------|
| 000000 | MDT | DSECT |
| 000000 | MDTNAME | DS CL32 |
| 000020 | MDTVERS | DS H |
| 000022 | MDTDATCR | DS CL8 |
| 00002A | MDTDATLU | DS CL8 |
| 00032 | MDTLEN | EQU *-MDT |

| | | | |
|--------|----------|-----------|---------------------------------|
| 000000 | MDT | DSECT | 17:41:48 02/18/83 |
| 000000 | MDTNAME | DS CL32 | NAME OF THE INCLUDED PROCESS |
| 000020 | MDTVERS | DS H | VERSION OF THE INCLUDED PROCESS |
| 000022 | MDTDATCR | DS CL8 | DATE CREATED |
| 00002A | MDTDATLU | DS CL8 | DATE LAST UPDATED |
| 00032 | MDTLEN | EQU *-MDT | LENGTH IN BYTES OF GMD |

1.85 #MSQDS

```
COPY #MSQDS
*****
***      MSQ:  ADS APPLICATION GENERATOR MENU SEQUENCE (INTERNAL) ***
***      MSQ IS A DSECT THAT IS THE INTERNAL CONTROL BLOCK USED TO ***
***      HOLD THE SEQUENCED RESPONSES FOR A MENU FUNCTION.
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-----------------------------|------------------------------------|
| 000000 | MSQ DSECT | 12:56:18 04/04/85 |
| 000000 | MSQNXTA DS A | ADDR OF NEXT MSQ |
| 000004 | MSQPREVA DS A | ADDR OF PREVIOUS MSQ |
| 000008 | MSQRECN DS H | PAGE NUMBER OF THIS ENTRY |
| 00000A | MSQCOUNT DS H | NUMBER OF RESPONSES IN THIS MSQ |
| 00000C | MSQHLEN EQU *-MSQ | LENGTH OF MSQ HEADER |
| 000003 | MSQHLENF EQU ((*-MSQ+3)/4) | LENGTH OF MSQ HEADER IN FULL WORDS |
| 00000C | MSQRESPS DS 0F | START OF RESPONSE NAME TABLE |

1.86 #MTEDS

```

COPY #MTEDS
*****
***      MTE IS THE DSECT THAT DEFINES THE MENU TABLE ELEMENT IN      ***
***      THE MENU TABLE POOL OF AN ADS APPLICATION DEFINITION      ***
***      BLOCK.                                                 ***
***      ***                                                 ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------------------|--|
| 000000 | MTE DSECT | 12:17:22 04/04/85 |
| | MTEHDR #FLAG X'80' | HEADER LINE TEXT MTE TYPE |
| 000000 | MTEHDRI DS 0XL1 | |
| 000080 | MTEHDRM EQU X'80' | |
| | MTERSQ #FLAG X'40' | SEQUENCED RESP MENU MTE TYPE |
| 000000 | MTERSQI DS 0XL1 | |
| 000040 | MTERSQM EQU X'40' | |
| 000000 | MTETYPE DS X | MTE TYPE FLAG |
| 000001 | DS X | UNUSED |
| | * | |
| | * HEADER LINE TEXT MTE LAYOUT | |
| | * | |
| 000002 | MTELIN# DS X | HEADING LINE NUMBER |
| 000003 | MTEHDLN DS X | LENGTH OF TEXT IN LITPOOL |
| 000004 | MTEHOFF DS F | LITPOOL OFFSET TO HEADER TEXT |
| 000008 | MTETEXT DS 0F | START OF TEXT *CRM84297* |
| 000008 | MTELEN EQU *-MTE | LENGTH OF MTE |
| 000002 | MTELENF EQU ((*-MTE+3)/4) | LENGTH OF MTE IN WORDS |
| | * | |
| | * | |
| | * SEQUENCED RESPONSE MENU MTE LAYOUT | |
| | * | |
| 000008 | ORG MTELIN# | |
| 000002 | MTENRSPS DS H | # OF RESP INDEXES FOLLOWING |
| 000004 | MTEHLEN EQU *-MTE | LENGTH OF RSQ MTE TYPE HEADER |
| 000001 | MTEHLENF EQU ((*-MTE+3)/4) | LENGTH OF RSQ MTE TYPE HEADER IN WORDS |
| 000004 | MTERSPS DS 0H | START OF RESP INDEX TABLE |

1.87 #MXDEDS

```

COPY #MXDEDS
*****
***      MXDE: MODEL XDE TABLE ENTRY
***      MXDE IS A DSECT THAT DESCRIBES AN ENTRY IN THE MODEL XDE
***      TABLE USED WHEN BUILT-IN FUNCTIONS ARE BEING PARSED. THIS
***      INFORMATION IS KEPT IN A TABLE IN ORDER TO ACCOMODATE
***      NESTED BUILT-IN FUNCTIONS.
***      LENGTH = OPND SPECIFIED
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|------------------------|----------------------------------|
| 000000 | MXDEDS DSECT | 15:36:24 06/29/84 |
| 000000 | MXDEADDR DS A | ADDR OF CURRENT MODEL XDE |
| 000004 | MXDEHdra DS A | ADDR OF CURR HDR MODEL XDE |
| 000008 | MXDEDATL DS H | DATA LENGTH |
| 00000A | MXDEDATP DS X | DATA TYPE |
| 00000B | MXDENODC DS X | DATA # OF DECIMALS |
| | MXDOPDL #FLAG X'80' | LENGTH = OPND SPECIFIED |
| 00000C | MXDOPDli DS 0XL1 | |
| 000080 | MXDOPDLM EQU X'80' | MXDCLCL #FLAG X'40' |
| | | LENGTH = CALC SPECIFIED |
| 00000C | MXDCLCLI DS 0XL1 | |
| 000040 | MXDCLCLM EQU X'40' | |
| | MXDOPDT #FLAG X'20' | DATATYPE=OPND SPECIFIED |
| 00000C | MXDOPDTI DS 0XL1 | |
| 000020 | MXDOPDTM EQU X'20' | |
| | MXDOPDC #FLAG X'10' | # OF DEC=OPND SPECIFIED |
| 00000C | MXDOPDCI DS 0XL1 | |
| 000010 | MXDOPDCM EQU X'10' | |
| | MXDVARP #FLAG X'08' | VARIABLE # OPERANDS SPECIFIED |
| 00000C | MXDVARPI DS 0XL1 | |
| 000008 | MXDVARPM EQU X'08' | |
| 00000C | MXDEFLG1 DS X | FLAG 1 |
| 00000D | DS XL3 | RESERVED |
| 00010 | MXDELEN EQU *-MXDEDS | LENGTH OF 1 ENTRY |
| 00004 | MXDELENF EQU MXDELEN/4 | LENGTH IN FULLWORDS OF ONE ENTRY |
| 00020 | MXDENENT EQU 32 | ARBITRARY # OF ENTRIES |

1.88 #MXSHDS

```

COPY #MXSHDS
*****
*      MXSHDS IS THE DSECT FOR A MOVED XDE STACK HEADER
*
*****
Offset  Value
000000  MXSHDR  DSECT          16:04:33 01/07/86
000000  MXSMILA DS  A          BASE ADDRESS FOR OTHER OFFSETS
000004  MXSMILIX DS  H          ACCUMULATOR
000006  MXSSFEOF DS  H          OFFSET TO CURRENT SFE
000008  MXSSDEOF DS  H          OFFSET TO CURRENT SDE
00000A  MXSSDECT DS  H          NUMBER OF SDE'S TO BE PROCESSED
00000C  MXSWORK  DS  H          WORK HALFWORD FOR ACCUM
00000E  MXSDXBDF DS  H          OFFSET TO CURRENT DXB (OR XDE)
000010  MXSBAKC  DS  A          BACK CHAIN TO PREVIOUS MOVED XDE STACK
000014  MXSPIKUP DS  A          PICKUP ADDRESS AFTER RESUMED SDE SCAN
000018  MXSXDERC DS  H          COUNT OF NUMBER OF XDE'S TO BE SCANNED
00001A  MXSMRELC DS  H          MRE LOOP COUNT
                  PRINT NOGEN
                  MXTRUE   #FLAG X'80'    TRUE FOUND INDICATOR
                  MXSFLSE  #FLAG X'40'    FALSE FOUND INDICATOR
                  MXSDMIL  #FLAG X'20'    OBTAIN MRE INDEX ONLY
00001C  MXSFLAG  DS  X          FLAG BYTE
00001D            DS  XL3
                  *** STANDARD XDE STACK HEADER BEGINS HERE ****
000020  MXSNXDE DS  H          NUMBER OF XDE'S AND DXB'S
000022  MXSIRASZ DS  H          IRA SIZE IN BYTES
000024  MXSSTKSZ DS  H          XDE STACK SIZE (NOT INCLUDING HDR)
                  MXSDTM  #FLAG X'20'    DATE/TIME REFERENCED IN STACK
                  PRINT GEN
000026  MXSMFLAG DS  X          FLAG BYTE          *LCB84258*
000027            DS  X          NUMBER OF USER FUNCTION XDE'S  **LCB84258*
00028  MXSFXDE EQU  *          FIRST XDE OR DXB IN XDE STACK
0000A  NRMXSWDS EQU  (MXSFXDE-MXSHDR)/4  NUMBER OF WORDS IN MOVED XDE
                  *          STACK HEADER
00008  SZXDESH EQU  MXSFXDE-MXSNXDE  NUMBER OF BYTES IN STANDARD XDE
                  *          STACK HEADER
00004  OFXDESS EQU  MXSSTKSZ-MXSNXDE  OFFSET (IN STANDARD STACK HDR) TO
                  *          XDE STACK SIZE FIELD
00002  OFNIRA   EQU  MXSIRASZ-MXSNXDE  OFFSET (IN STANDARD STACK HDR) TO
                  *          NUMBER OF IRA BYTES FIELD
00007  OFUFCT   EQU  MXSUFNCT-MXSNXDE  OFFSET (IN STANDARD STACK*JEB84145*
                  *          HDR) TO # USER FUNCS  *JEB84145*

```

1.89 #NAMEDS

```
COPY #NAMEDS
*****
**          DSECT FOR DRN NAME LIST.
**
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-----------------------------------|--------------------------------|
| 000000 | NAMEDS DSECT | 12:07:02 04/04/85 |
| 000000 | NAMENEXT DS A | * ADDRESS OF NEXT NAME IN LIST |
| 000004 | NAMEKEY DS 0CL10 | * LENGTH OF KEY |
| 000004 | NAMEVALU DS CL8 | * VALUE OF NAME |
| 00000C | NAMEVERS DS H | * VALUE OF VERSION |
| 00010 | NAMELEN EQU ((*-NAMEDS+3)/4*4) | * LENGTH OF THIS DSECT |

1.90 #OCBDS

```

COPY #OCBDS
*****
***          OCB: ADSO CONTROL BLOCK DSECT
***          OCB IS A DSECT THAT DESCRIBES THE ADSO CONTROL BLOCK
***          USED BY THE ADSO RUNTIME SYSTEM. IT CONTAINS SYSTEM
***          WIDE GLOBAL INFORMATION USED BY ALL ADSO TASKS. IT
***          IS BUILT AT ADSO STARTUP TIME FROM SYSGEN DATA STORED
***          ON THE DICTIONARY.
***          ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------|-------------------------------------|
| 000000 | OCB DSECT | 09:20:04 04/29/86 10/17/90 |
| 000000 | OCBID DS CL4 | "OCB*" |
| 000004 | OCBMNTC DS CL8 | MENU PROCESSING PROGRAM TASK CODE |
| 00000C | OCBMANTC DS CL8 | MAIN PROCESSOR TASK CODE |
| 000014 | OCBAPRNM DS CL8 | AUTO DIALOG NAME |
| 00001C | OCBPBFSZ DS F | SIZE OF PRIMARY REC BUFF POOL |
| 000020 | OCBSBFSZ DS F | SIZE OF SECONDARY REC BUFF POOL |
| 000024 | OCBTHRSH DS F | THRESHOLD FOR SCRATCH STORAGE |
| 000028 | OCBA2UTC DS CL8 | INTERNAL TASK CODE UNDER UMBR(TCF) |
| 000030 | DS F | RESERVED FOR FUTURE USE |
| 000034 | OCBMXLNK DS H | MAX LINK LEVEL |
| | *OCBUMEN #FLAG X'80' | ON IF USER ONLY MENU |
| | *OCBMNKP #FLAG X'40' | ON IF MENU TO BE KEPT IN SCRATCH |
| | *OCBAST #FLAG X'20' | ON IF AUTOSTATUS = YES |
| | *OCBASTM #FLAG X'10' | ON IF AUTOSTATUS IS MANDATORY |
| | *OCBSDRM #FLAG X'08' | ON IF STATUS DEFN REC MANDATORY |
| | *OCBDIAG #FLAG X'04' | ON IF BYPASS DIAGNOSTIC SCREEN SET |
| | *OCBNWP #FLAG X'02' | ON IF MAPOUT NEWPAGE REQUIRED |
| | *OCBRELO #FLAG X'01' | ON IF STORAGE IS TO BE RELOCATABLE |
| | *OCBFLAG1 DS XL1'00' | FLAG BYTE 1 |
| | *OCBACT #FLAG X'80' | ON IF ACTIVITY LOGGING SET TO YES |
| | *OCBSTAT #FLAG X'40' | ON IF DIALOG STATS BEING COLLECTED |
| | *OCBCSEL #FLAG X'20' | ON IF SELECTED DIALOG STATS |
| | *OCBCOBM #FLAG X'10' | ON IF COBOL MOVE SET TO YES 0* |
| | *OCBSMCL #FLAG X'08' | ON IF STORAGE MODE IS CALCULATED 0* |
| | *OCBSMCM #FLAG X'04' | ON IF STORAGE MODE IS COMPRESSED 0* |
| | *OCBCBLM #FLAG X'02' | ON IF COBOL MOVE SETTING MANDATORY |
| | *OCBFLAG2 DS XL1'00' | FLAG BYTE 2 |
| | PRINT NOGEN | |
| | OCBUMEN #FLAG X'80' | ON IF USER ONLY MENU |
| | OCBMNKP #FLAG X'40' | ON IF MENU TO BE KEPT IN SCRATCH |
| | OCBAST #FLAG X'20' | ON IF AUTOSTATUS = YES |
| | OCBASTM #FLAG X'10' | ON IF AUTOSTATUS IS MANDATORY |
| | OCBSDRM #FLAG X'08' | ON IF STATUS DEFN REC MANDATORY |
| | OCBDIAG #FLAG X'04' | ON IF BYPASS DIAGNOSTIC SCREEN SET |
| | OCBNWP #FLAG X'02' | ON IF MAPOUT NEWPAGE REQUIRED |
| | OCBRELO #FLAG X'01' | ON IF STORAGE IS TO BE RELOCATABLE |
| 000036 | OCBFLAG1 DS XL1'00' | FLAG BYTE 1 |
| | OCBACT #FLAG X'80' | ON IF ACTIVITY LOGGING SET TO YES |
| | OCBSTAT #FLAG X'40' | ON IF DIALOG STATS BEING COLLECTED |
| | OCBCSEL #FLAG X'20' | ON IF SELECTED DIALOG STATS |
| | OCBCOBM #FLAG X'10' | ON IF COBOL MOVE SET TO YES 0* |
| | OCBSMCL #FLAG X'08' | ON IF STORAGE MODE IS CALCULATED |
| | OCBSMCM #FLAG X'04' | ON IF STORAGE MODE IS COMPRESSED |
| | OCBCBLM #FLAG X'02' | ON IF COBOL MOVE SETTING MANDATORY |
| 000037 | OCBFLAG2 DS XL1'00' | FLAG BYTE 2 |

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------|------------------------------------|
| 000000 | OCB DSECT | 09:20:04 04/29/86 10/17/90 |
| 000000 | OCBID DS CL4 | "OCB*" |
| 000004 | OCBMNTC DS CL8 | MENU PROCESSING PROGRAM TASK CODE |
| 00000C | OCBMANTC DS CL8 | MAIN PROCESSOR TASK CODE |
| 000014 | OCBAPRNM DS CL8 | AUTO DIALOG NAME |
| 00001C | OCBPBFSZ DS F | SIZE OF PRIMARY REC BUFF POOL |
| 000020 | OCBSBFSZ DS F | SIZE OF SECONDARY REC BUFF POOL |
| 000024 | OCBTHRSH DS F | THRESHOLD FOR SCRATCH STORAGE |
| 000028 | OCBA2UTC DS CL8 | INTERNAL TASK CODE UNDER UMBR(TCF) |
| 000030 | DS F | RESERVED FOR FUTURE USE |
| 000034 | OCBMXLNK DS H | MAX LINK LEVEL |
| | *OCBUMEN #FLAG X'80' | ON IF USER ONLY MENU |
| | *OCBMNKP #FLAG X'40' | ON IF MENU TO BE KEPT IN SCRATCH |
| | *OCBAST #FLAG X'20' | ON IF AUTOSTATUS = YES |
| | *OCBASTM #FLAG X'10' | ON IF AUTOSTATUS IS MANDATORY |
| | *OCBSDRM #FLAG X'08' | ON IF STATUS DEFN REC MANDATORY |
| | *OCBDIAG #FLAG X'04' | ON IF BYPASS DIAGNOSTIC SCREEN SET |
| | *OCBNWP #FLAG X'02' | ON IF MAPOUT NEWPAGE REQUIRED |
| | *OCBRELO #FLAG X'01' | ON IF STORAGE IS TO BE RELOCATABLE |
| | *OCBFLAG1 DS XL1'00' | FLAG BYTE 1 |
| | *OCBACT #FLAG X'80' | ON IF ACTIVITY LOGGING SET TO YES |
| | *OCBSTAT #FLAG X'40' | ON IF DIALOG STATS BEING COLLECTED |
| | *OCBCSEL #FLAG X'20' | ON IF SELECTED DIALOG STATS |
| | *OCBCOBM #FLAG X'10' | ON IF COBOL MOVE SET TO YES 0* |
| | *OCBSMCL #FLAG X'08' | ON IF STORAGE MODE IS CALCULATED |
| | *OCBSMCM #FLAG X'04' | ON IF STORAGE MODE IS COMPRESSED |
| | *OCBCBLM #FLAG X'02' | ON IF COBOL MOVE SETTING MANDATORY |
| | OCBFLAG2 DS XL1'00' | FLAG BYTE 2 |
| | PRINT NOGEN | |
| | OCBUMEN #FLAG X'80' | ON IF USER ONLY MENU |
| | OCBMNKP #FLAG X'40' | ON IF MENU TO BE KEPT IN SCRATCH |
| | OCBAST #FLAG X'20' | ON IF AUTOSTATUS = YES |
| | OCBASTM #FLAG X'10' | ON IF AUTOSTATUS IS MANDATORY |
| | OCBSDRM #FLAG X'08' | ON IF STATUS DEFN REC MANDATORY |
| | OCBDIAG #FLAG X'04' | ON IF BYPASS DIAGNOSTIC SCREEN SET |
| | OCBNWP #FLAG X'02' | ON IF MAPOUT NEWPAGE REQUIRED |
| | OCBRELO #FLAG X'01' | ON IF STORAGE IS TO BE RELOCATABLE |
| 000036 | OCBFLAG1 DS XL1'00' | FLAG BYTE 1 |
| | OCBACT #FLAG X'80' | ON IF ACTIVITY LOGGING SET TO YES |
| | OCBSTAT #FLAG X'40' | ON IF DIALOG STATS BEING COLLECTED |
| | OCBCSEL #FLAG X'20' | ON IF SELECTED DIALOG STATS |
| | OCBCOBM #FLAG X'10' | ON IF COBOL MOVE SET TO YES 0* |
| | OCBSMCL #FLAG X'08' | ON IF STORAGE MODE IS CALCULATED |
| | OCBSMCM #FLAG X'04' | ON IF STORAGE MODE IS COMPRESSED |
| | OCBCBLM #FLAG X'02' | ON IF COBOL MOVE SETTING MANDATORY |
| 000037 | OCBFLAG2 DS XL1'00' | FLAG BYTE 2 |

```

        PRINT GEN
000038    OCBSDRV DS H           STATUS DEF REC VERSION
00003A    OCBCKPT DS H           STATISTICS CHECKPOINT COUNT
00003C    OCBSDRNM DS CL32      STATUS DEF REC NAME
0005C     OCBLEN EQU *-OCB      LENGTH OF DSECT
00005C    OCBRTCDE DS H          ROUTE CODE FLAGS
00005E    OCBDSCDE DS H          DESCRIPTOR CODE FLAGS
000060    OCBDMCNT DS H          DIALOG MAXIMUM ERROR COUNT
000062    OCBMXLRF DS H          LRF PACKET STACK MAX SIZE*PHH86118*
000064    OCBMXQUE DS H          QUEUE/PRINTER MAX REC LEN*PHH86118*
*OCBROAU #FLAG X'80'          ON IF RESTART OPTION AUTOMATIC
*OCBROPR #FLAG X'40'          ON IF RESTART OPTION PROMPT
*OCBROMA #FLAG X'20'          ON IF RESTART OPTION MANUAL
*OCBCHIG #FLAG X'10'          ON IF CHECKPOINTS IGNORED
*OCBOPSD #FLAG X'08'          OPERATOR SHUTDOWN ENABLED
*OCBCOMN #FLAG X'04'          COMMIT CAUSES NOACTION
*OCBCOMW #FLAG X'02'          COMMIT CAUSES WARNING MSGS
*OCBCOMA #FLAG X'01'          COMMIT CAUSES ABEND
PRINT NOGEN
OCBROAU #FLAG X'80'          ON IF RESTART OPTION AUTOMATIC
OCBROPR #FLAG X'40'          ON IF RESTART OPTION PROMPT
OCBROMA #FLAG X'20'          ON IF RESTART OPTION MANUAL
OCBCHIG #FLAG X'10'          ON IF CHECKPOINTS IGNORED
OCBOPSD #FLAG X'08'          OPERATOR SHUTDOWN ENABLED
OCBCOMN #FLAG X'04'          COMMIT CAUSES NOACTION
OCBCOMW #FLAG X'02'          COMMIT CAUSES WARNING MSGS
OCBCOMA #FLAG X'01'          COMMIT CAUSES ABEND
PRINT GEN
*
*** IF THE ABOVE THREE FLAGS (OCBROAU, OCBROPR AND OCBROMA) ARE ALL
*** OFF, THEN RESTART OPTION IS "NO".
*
000066    OCBFLAGB DS XL1         BATCH FLAG BYTES
000067    DS XL1                 RESERVED
000068    OCBLOB DS 0C            BEGINING OF BATCH LOB AREA
000068    DS CL60                (AS OF 07/15/85, ONLY 52 BYTES ARE
*                                NEEDED FOR THE LOG OPTIONS BLOCK)
0000A4    DS F                  RESERVED
000A8    OCBBLEN EQU *-OCB      LENGTH OF DSECT FOR BATCH

```

1.91 #OCCLDS

COPY #OCCLDS

| <u>Offset</u> | <u>Value</u> | | | |
|---------------|-----------------------|--------------|-------------------------------------|--|
| 000000 | OCCLDS | DSECT | 08:58:05 01/31/86 | |
| 000000 | 5CD6C3C3D3C3C250CCLID | DC | CL8'*OCCLCB*' DUMP ID | |
| 000008 | OCCL83DK | DS A | DBKEY OF NAMESYN-083 (INPUT) | |
| 00000C | OCCLSSCA | DS A | ADDR OF SUBSCHEMA CTL (INPUT) | |
| 000010 | OCCL42BA | DS A | SDR-042 RECORD BUFFER ADDRESS | |
| 000014 | OCCLFDOF | DS H | OFFSET TO BE FIRST OCCURRENCE | |
| 000016 | | DS H | RESERVED | |
| 000018 | | DS 2F | RESERVED | |
| 000020 | OCCLRC | DS F | RETURN CODE | |
| | * | | 0 - HOME FREE | |
| | * | | 4 - ELEMENT DOES NOT OCCUR | |
| | * | | 8 - NOT ENOUGH RECORD STRUCTURE | |
| | * | | 12 - FATAL DATABASE OR LOGIC ERROR | |
| 000024 | OCCL#LVS | DS H | NUMBER OF SUBSCRIPT LEVELS (OUTPUT) | |
| 000026 | OCCLOCC1 | DS H | # OF OCCURANCES: LEVEL 1 (OUTPUT) | |
| 000028 | OCCLOFF1 | DS H | LENGTH OF ENTIRE GROUP: 1 (OUTPUT) | |
| 00002A | OCCLOCC2 | DS H | # OF OCCURANCES: LEVEL 2 (OUTPUT) | |
| 00002C | OCCLOFF2 | DS H | LENGTH OF ENTIRE GROUP: 2 (OUTPUT) | |
| 00002E | OCCLOCC3 | DS H | # OF OCCURANCES: LEVEL 3 (OUTPUT) | |
| 000030 | OCCLOFF3 | DS H | LENGTH OF ENTIRE GROUP: 3 (OUTPUT) | |
| 000032 | OCCLOCC4 | DS H | # OF OCCURANCES: LEVEL 4 (OUTPUT) | |
| 000034 | OCCLOFF4 | DS H | LENGTH OF ENTIRE GROUP: 4 (OUTPUT) | |
| 000036 | OCCLOCC5 | DS H | # OF OCCURANCES: LEVEL 5 (OUTPUT) | |
| 000038 | OCCLOFF5 | DS H | LENGTH OF ENTIRE GROUP: 5 (OUTPUT) | |
| 00003A | OCCLOCC6 | DS H | # OF OCCURANCES: LEVEL 6 (OUTPUT) | |
| 00003C | OCCLOFF6 | DS H | LENGTH OF ENTIRE GROUP: 6 (OUTPUT) | |
| 00003E | OCCLOCC7 | DS H | # OF OCCURANCES: LEVEL 7 (OUTPUT) | |
| 000040 | OCCLOFF7 | DS H | LENGTH OF ENTIRE GROUP: 7 (OUTPUT) | |
| 000042 | OCCLOCC8 | DS H | # OF OCCURANCES: LEVEL 8 (OUTPUT) | |
| 000044 | OCCLOFF8 | DS H | LENGTH OF ENTIRE GROUP: 8 (OUTPUT) | |
| 000046 | OCCLOCC9 | DS H | # OF OCCURANCES: LEVEL 9 (OUTPUT) | |
| 000048 | OCCLOFF9 | DS H | LENGTH OF ENTIRE GROUP: 9 (OUTPUT) | |
| 00004A | OCCLOCCA | DS H | # OF OCCURANCES: LEVEL 10 (OUTPUT) | |
| 00004C | OCCLOFFA | DS H | LENGTH OF ENTIRE GROUP: 10 (OUTPUT) | |
| 00004E | OCCLOCCB | DS H | # OF OCCURANCES: LEVEL 11 (OUTPUT) | |
| 000050 | OCCLOFFB | DS H | LENGTH OF ENTIRE GROUP: 11 (OUTPUT) | |
| 000052 | OCCLOCCC | DS H | # OF OCCURANCES: LEVEL 12 (OUTPUT) | |
| 000054 | OCCLOFFC | DS H | LENGTH OF ENTIRE GROUP: 12 (OUTPUT) | |
| 000056 | OCCLOCCD | DS H | # OF OCCURANCES: LEVEL 13 (OUTPUT) | |
| 000058 | OCCLOFFD | DS H | LENGTH OF ENTIRE GROUP: 13 (OUTPUT) | |
| 00005A | OCCLOCCE | DS H | # OF OCCURANCES: LEVEL 14 (OUTPUT) | |
| 00005C | OCCLOFFE | DS H | LENGTH OF ENTIRE GROUP: 14 (OUTPUT) | |
| 00005E | OCCLOCCF | DS H | # OF OCCURANCES: LEVEL 15 (OUTPUT) | |
| 000060 | OCCLOFFF | DS H | LENGTH OF ENTIRE GROUP: 15 (OUTPUT) | |
| 00062 | OCCLEN | EQU *-OCCLDS | | |

1.92 #OFDDS

```

COPY #OFDDS
*****
***          #OFDDS
***  OFD DESCRIBES THE OCCURRING FIELDS DESCRIPTOR USED BY THE
***  ADS/ONLINE GENERATOR TO CONTROL THE INITIALIZATION OF
***  OCCURRING FIELDS.
***          ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------------|-----------------------------------|
| 000000 | OFD DSECT | **R14.1** |
| 000000 | OFDNEXTA DS A | ADDRESS OF NEXT OFD ENTRY |
| 000004 | OFDPREVA DS A | ADDRESS OF PRIOR OFD ENTRY |
| 000008 | OFDRECOF DS H | FIRST OFFSET WITHIN GROUP |
| 00000A | OFDRCBOF DS H | FIRST GROUP BIT OFFSET*LBCB84109* |
| 00000C | OFDCUROF DS H | CURRENT OFFSET WITHIN GROUP |
| 00000E | OFDCRBOF DS H | CURR GROUP BIT OFFSET *LBCB84109* |
| 000010 | OFDMAXOC DS H | MAXIMUM OCCURRENCES OF GROUP |
| 000012 | OFDCUROC DS H | CURRENT OCCURRENCE LEVEL |
| 000014 | OFDGSIZE DS H | SIZE OF THIS OCCURRING GROUP |
| | OFDMBB #FLAG X'80' | MULTIBIT BINARY FIELD *LCB84109* |
| 000016 | OFDMBBI DS 0XL1 | |
| 00080 | OFDMBBM EQU X'80' | |
| 000016 | OFDFLAG1 DS X | FLAG FIELD *LCB84109* |
| 000017 | DS X | RESERVED *LCB84109* |
| 00006 | OFDLENF EQU ((*-OFD+3)/4) | LENGTH OF OFD IN WORDS |

1.93 #OFTDS

```
COPY #OFTDS
*****
***   OFT IS THE DSECT FOR THE OFFSET TABLE FOUND AS PART OF THE ***
***   PME/RSE STRUCTURE FOUND IN THE FDB. IT IS POINTED TO BY THE ***
***   PME/RSE AND IS USED BOTH FOR REASSESSING THE OFFSET OF TRUE/ ***
***   FALSE CLAUSES IN THE CME STRUCTURE (FOR THE NEW EXECUTABLE ***
***   CODE FDB'S) AS WELL AS FOR REPORTING ON THE CONTENTS OF THE ***
***   FDB WHEN ADSORPTS IS BEING USED.
***   ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------|
| 000000 | OFT | DSECT |
| 000000 | OFTCLAS | DS H |
| 000002 | OFTFUNC | DS H |
| 000004 | OFTOFF | DS F |
| 000008 | OFTNOFF | DS F |
| 0000C | OFTLEN | EQU ((*-OFT)/4)*4 |
| 00003 | OFTLENF | EQU ((*-OFT+3)/4) |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-------------------|
| 000000 | OFT | DSECT |
| 000000 | OFTCLAS | DS H |
| 000002 | OFTFUNC | DS H |
| 000004 | OFTOFF | DS F |
| 000008 | OFTNOFF | DS F |
| 0000C | OFTLEN | EQU ((*-OFT)/4)*4 |
| 00003 | OFTLENF | EQU ((*-OFT+3)/4) |

16:03:10 09/14/83

CME CLASS

CME FUNCTION

OFFSET TO OLD CME

OFFSET TO NEW CME

LENGTH OF OFT FIXED PORTION

OF FULLWORDS IN FIXED PORTION

1.94 #OTBXDS

```

#OTBXDS
*****
***      OTBEXT: EXTENSION IN THE ADSO TERMINAL BLOCK (OTB)      ***
***      FOR APPLICATIONS DEFINED BY ADSA.                         ***
***      OTBEXT CONTAINS DATA ELEMENTS AND ADDRESSES REQUIRED BY   ***
***      THE ADS RUNTIME SYSTEM WHEN EXECUTING APPLICATIONS DEFINED   ***
***      BY THE APPLICATION GENERATOR (ADSA).                         ***
***      THIS EXPANSION OF OTBEXT IS FOR PROGRAMS WHICH WILL RUN    ***
***      IN BOTH THE ONLINE AND BATCH ENVIRONMENTS.                 ***
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> |
|---------------|--|
| 000000 | OTBEXT DSECT |
| 000000 | OTBEXTID DS CL4'OTX*' EYE CATCHER 0 |
| 000004 | OTBXADBA DS A(0) ADDRESS OF THE APPL DEFN BLOCK (ADB) |
| 000008 | OTBXTATA DS A(0) ADDRESS OF THE TASK APPL TABLE (TAT) |
| 00000C | OTBXTFEA DS F'0' OFFSET TO TASK/TOP FUNC ENTRY (TFE) |
| 000010 | OTBXFDEA DS F'0' OFFSET TO FUNC DEFN ELEMENT (FDE) |
| 000014 | OTBXMENA DS A *ONLINE* ADDRESS OF THE MENU STACK |
| 000018 | ORG OTBXMENA |
| 000014 | DS F'0' *BATCH* RESERVED |
| 000018 | OTBXAPGA DS A(0) ADDRESS OF APPL GLOBAL RECORD (APG) |
| * | NOTE: POSITION OF OTBXAPGA CANNOT |
| * | . CHANGE BECAUSE RUNTIME EVALUATOR |
| * | . (RTE) EXPECTS IT AT +X'18' *PHH86120* |
| 00001C | OTBXAPRA DS F'0' OFFSET TO RESPONSE ELEMENT (APRE) |
| 000020 | OTBXADBL DS F'0' LENGTH OF THE ADB |
| 000024 | OTBXSVRA DS F'0' REG SAVE AREA (ACF BITS PROCESS CNTR) |
| 000028 | OTBXSVRB DS F'0' REGISTER SAVE AREA |
| 00002C | OTBXSVRC DS F'0' REGISTER SAVE AREA |
| 000030 | OTBXTPMA DS A *ONLINE* ADDRESS OF TOP MENU STACK ENTRY |
| 000034 | ORG OTBXTPMA |
| 000030 | DS F'0' *BATCH* RESERVED |
| 000034 | DS F RESERVED |
| 000038 | OTBXHLMA DS A *ONLINE* ADDR OF ACF HELP MAP (MRB AND RECORD) |
| 00003C | OTBXMNRA DS A *ONLINE* ADDRESS OF ACF MENU RECORD |
| 000040 | OTBXRTE DS F *ONLINE* CURRENT MENU RESPONSE TABLE ENTRY ADDR |
| 000044 | ORG OTBXHLMA |
| 000038 | DS 3F'0' *BATCH* RESERVED |
| 000044 | OTBXTAEN DS CL8' ' APPLICATION NAME |
| 00004C | OTBXTAEV DS H'0' APPLICATION VERSION |
| 00004E | OTBXFAPR DS H *ONLINE* FIRST APRE FOR A REQUESTED MENU PAGE |
| 000050 | OTBXTAPR DS H *ONLINE* TOTAL APRE QUALIFYING FOR MAP |
| 000052 | OTBXMSQN DS H *ONLINE* # MENU SEQUENCE RESPONSES IN MTE |
| 000054 | OTBXPGR DS X *ONLINE* COUNT OF APRE ON REQUESTED MENU PAGE |
| * OTBXSDC | #FLAG X'80' *ONLINE* USER IS SIGNED ON TO DC |
| * OTBXHLP | #FLAG X'40' *ONLINE* RETURN TO RUN2 FOR HELP SCREEN |
| * OTBXAPR | #FLAG X'20' *ONLINE* DETERMINE QUALIFYING NUMBER OF APRE'S |
| * OTBXRSQ | #FLAG X'10' *ONLINE* GET MENU RESPONSES IN SEQUENCE |
| | PRINT NOGEN |
| | PRINT GEN |
| 000055 | OTBXFLAG DS X *ONLINE* FIRST FLAG BYTE |
| 000056 | ORG OTBXFAPR |
| 00004E | DS 4H'0' *BATCH* RESERVED |
| * OTBXSOS | #FLAG X'80' USER IS SIGNED ON TO THE APPLICATION |
| * OTBXAGR | #FLAG X'40' ALLOCATING ONLY THE APPL GLOBAL RECS |
| * OTBXGR1 | #FLAG X'20' APPLICATION GLOBAL RECORD INITIALIZED |

```
* OTBXNVL  #FLAG X'10'      INVALID RESPONSE FOR APPLICATION
* OTBXNPR  #FLAG X'08'      INVALID RESPONSE SET BY PROCESS
* OTBXRCH  #FLAG X'04'      NON-ENTER RKE'S CHECKED; NONE VALID
* OTBXVRF  #FLAG X'02'      VALID RKE FOUND          88-05-1119
      PRINT NOGEN
      PRINT GEN
000056    OTBXFLG2 DS   X'00'      2ND FLAG BYTE
000057        DS   X           RESERVED
*** Change for PTF# 87-05-1055.      MCM
000058    OTBSTAMP DS   CL24      Used to save ADB name,date,time.  MCM
000070    OTBXBMPA DS   A(0)      SECURITY BIT MAP ADDRESS      GWG
000074        DS   3F           RESERVED
00080    OTBXLEN EQU   *-OTBEXT LENGTH OF OTB EXTENSION
*
00078    OTBXSBN EQU   120      *ONLINE* LENGTH OF MENU STACK (20 ENTRIES MAX)
00020    BITMAPL EQU   32       LENGTH OF BIT MAP (256 BITS) *ADS370/194*
00002   SECLNTRY EQU   2        LENGTH OF 1 SEC CLASS PARM *ADS370/194*
00200   SECLISTL EQU   SECLNTRY*256 LENGTH OF SEC CLASS PARM LST *ADS370/194*
```

1.95 #PEXCDS

```

COPY #PEXCDS
*****
*** PEXC IS THE DSECT USED FOR CHAINING THE EXECUTABLE CODE ***
*** DSECTS TOGETHER IN A LOGICAL SEQUENCE. THERE WILL BE ONE ***
*** PEXC PER PROCESS MUCH THE SAME WAY THERE IS ONE GPR PER ***
*** PROCESS IN THE ADSO GENERATOR CONTROL BLOCKS. ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-----------------------------|-----------------------------------|
| 000000 | PEXC DSECT | 14:35:47 04/02/84 |
| 000000 | PEXCID DS CL4 | 'PEXC' EYE-CATCHER |
| 000004 | PEXCNTA DS F | ADDR OF NEXT PEXC IN CHAIN |
| 000008 | PEXCA DS F | POINTER TO FIRST EXC ON CHAIN |
| 00000C | PEXCCA DS F | POINTER TO CURRENT EXC ON CHAIN |
| 000010 | PEXCFOFT DS F | 1ST VALUE IN OFFSET TABLE |
| 000014 | PEXCCOFT DS F | CURRENT VALUE IN OFFSET TABLE |
| 000018 | PEXCFCME DS F | FIRST CME ADDRESS |
| 00001C | PEXLNTA DS F | LINE # TABLE ADDRESS |
| 000020 | PEXCNAM DS CL32 | PROCESS NAME (PREMAP OR RESPONSE) |
| 000040 | PEXCVER DS H | PROCESS VERSION |
| | PEXCRDP #FLAG X'80' | THIS IS A DUPLICATE RESPONSE |
| 000042 | PEXCRDPI DS 0XL1 | |
| 000080 | PEXCRDPM EQU X'80' | |
| 000042 | PEXCFLG1 DS X | FLAG BYTE |
| 00040 | PEXCLEN EQU ((*-PEXC)/4)*4 | LENGTH OF PEXC FIXED PORTION |
| 00011 | PEXCLENF EQU ((*-PEXC+3)/4) | # OF FULLWORDS IN FIXED PORTION |

1.96 #PFLDS

```
COPY #PFLDS
*****
*** PFLDS IS A DSECT FOR DESCRIBING THE PFL AREA IN MEMORY, ***
*** WHICH EVENTUALLY TRANSLATED INTO THE PANELFLD-121 AND ***
*** PFLD-DATA-147 RECORDS ON THE DATA DIRECTORY, USED BY THE ***
*** MAPPING COMPILER. THIS AREA CONSISTS OF A FIXED HEADER ***
*** PORTION, A SERIES OF DEVICE DEPENDENT PORTIONS AND A ***
*** LITERAL POOL AT THE END.
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|------------------------------------|
| 000000 | PFL DSECT | 04:34:44 05/22/86 |
| | PRINT NOGEN | |
| 000000 | PFLNAME DS CL32 | PANEL FIELD NAME |
| 000020 | PFLTOTLN DS H | TOTAL LENGTH OF AREA |
| 000022 | PFLOCCUR DS H | OCCURS VALUE |
| | PFLVALU #FLAG X'80' | PFLD IS A LITERAL VALUE |
| 000024 | PFLFLG1 DS X | FLAG BYTE |
| 000025 | PFLDVCNT DS X | NUMBER OF DEVICE GROUPS |
| | PFLPSD #FLAG X'80' | START OF DETAIL |
| | PFLPED #FLAG X'40' | END OF DETAIL |
| | PFLPOD #FLAG X'C0' | ONLY DETAIL DATA FIELD |
| | PFLPFS #FLAG X'20' | START OF FOOTER |
| | PFLPGA #FLAG X'E0' | ALL PAGING BITS |
| 000026 | PFLPGFLG DS X | PAGABLE MAP FLAG BYTE |
| 000027 | PFLFIL1 DS CL1 | RESERVED |
| 00028 | PFLBASLN EQU *-PFL | |
| | *** THE DEVICE DEPENDENT SECTION MAY ITERATE UP TO THE NUMBER OF | |
| | *** DEVICES SPECIFIED IN THE PANEL. IT CONSIST OF A DEVICE FOUNDATION | |
| | *** AND A SERIES OF ROW/COLUMN PAIRS, CORRESPONDING TO THE OCCURS | |
| | *** VALUE, PFLOCCUR. | |
| 000028 | PFLDEVIC DS 0F | |
| 000028 | PFLNXTA DS A | OFFSET OF NEXT DEVICE SECTION |
| 00002C | PFLLITPL DS A | OFFSET OF THIS DEVICE'S LITPOOL |
| 000030 | PFLLITLN DS H | LENGTH OF DEVICE'S LITERAL POOL |
| | PFL1240 #FLAG X'80' | 12 BY 40 SCREEN |
| | PFL1280 #FLAG X'40' | 12 BY 80 SCREEN |
| | PFL2480 #FLAG X'20' | 24 BY 80 SCREEN |
| | PFL3280 #FLAG X'10' | 32 BY 80 SCREEN |
| | PFL4380 #FLAG X'08' | 43 BY 80 SCREEN |
| | PFL2713 #FLAG X'02' | 27 BY 132 SCREEN |
| 000032 | PFLDEVS DS X | DEVICES DEFINEDBY THIS ENTRY |
| | PFLDLIM #FLAG X'80' | DELIMIT = YES |
| | PFLDLMN #FLAG X'40' | DELIMIT, WITH NOSKIP |
| 000033 | PFLDFLG DS X | GENERAL FLAG BYTE FOR DEVICE ENTRY |
| | *** 3270 ATTRIBUTE BYTE | |
| | PFLPROT #FLAG X'20' | PROTECTED |
| | PFLNUM #FLAG X'10' | NUMERIC |
| | PFLNDIS #FLAG X'0C' | NON-DISPLAY/NON-DETECTABLE |
| | PFLBRIT #FLAG X'08' | BRIGHT/DETECTABLE |
| | PFLDISD #FLAG X'04' | DISPLAY/DETECTABLE |
| | PFLMDT #FLAG X'01' | MODIFY DATA TAG |
| 000034 | PFLATTR DS X | |
| | PFLLEFT #FLAG X'08' | LEFT OUTLINING |
| | PFLTOP #FLAG X'04' | TOP OUTLINING |
| | PFLRGHT #FLAG X'02' | RIGHT OUTLINING |
| | PFLBOT #FLAG X'01' | BOTTOM OUTLINING |
| | PFLALL #FLAG X'0F' | ALL OUTLINING |
| 000035 | PFLATTR4 DS X | OUTLINING SUPPORT |
| | *** ATTRIBUTE BYTE FOR EXTENDED HIGHLIGHTING(3279S) | |
| | PFL2DEF #FLAG X'00' | SELECT DEFAULT |
| | | BZJ |

```
PFL2BLK #FLAG X'F1'          BLINK
PFL2REV #FLAG X'F2'          REVERSE-VIDEO
PFL2USC #FLAG X'F4'          UNDERSCORE
000036 PFLATTR2 DS   X
*** ATTRIBUTE BYTE FOR EXTENDED COLOUR(3279S)
PFL3DEF #FLAG X'00'          SELECT DEFAULT
PFL3BLU #FLAG X'F1'          BLUE
PFL3RED #FLAG X'F2'          RED
PFL3PIN #FLAG X'F3'          PINK
PFL3GRE #FLAG X'F4'          GREEN
PFL3TUR #FLAG X'F5'          TURQUOISE
PFL3YEL #FLAG X'F6'          YELLOW
PFL3WHI #FLAG X'F7'          WHITE
000037 PFLATTR3 DS   X
00010 PFLDEVLN EQU  *--PFLDEVIC LENGTH OF DEVICE DEPENDANT BASE
*** EACH DEVICE BASE IS FOLLOWED IMMEDIATELY BY N ROW/COLUMN PAIRS,
*** WHERE N = OCCURS VALUE (PFLOCCUR).
000038 PFLCOORD DS  OH
000038 PFLROW  DS   X          FIELD'S ROW POSITION
000039 PFLCOLM DS   X          FIELD'S COLUMN POSITION
*** AT END OF THE DEVICE DEPENDANT SECTION, A LITERAL POOL FOLLOWS,
*** CONSISTING OF ALL LITERALS SPECIFIED IN VALUE CLAUSES FOR ALL
*** OCCURENCES OF THIS PANEL FIELD. EACH LITERAL IS PRECEDED BY A
** 2 BYTE ITERATION FACTOR AND A 2 BYTE LENGTH.
00003A PFLLIT  DS   0X
00003A PFLITER DS  XL2          # OF ITERATIONS FOR THIS VALUE
00003C PFLLEN   DS  XL2          LENGTH OF THIS LITERAL
00003E PFLLITRL DS   0X          START OF LITERAL VALUE
PRINT GEN
```

1.97 #PICDS

```

COPY #PICDS
*****
*** PIC IS THE DSECT USED TO DEFINED THE AREA TO BE USED FOR ***
*** HOLDING THE PICTURE OF THE DISPLAY TARGET OF AN EDITED MOVE. ***
*** ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------------|--------------------------------|
| 000000 | PIC DSECT | 12:47:45 05/15/84 |
| | PICBONZ #FLAG X'80' | BLANK ON ZERO CONDITION |
| 000000 | PICBONZI DS 0XL1 | |
| 000080 | PICBONZM EQU X'80' | |
| | PICJUST #FLAG X'40' | JUSTIFY CONDITION |
| 000000 | PICJUSTI DS 0XL1 | |
| 000040 | PICJUSTM EQU X'40' | |
| 000000 | PICFLAG1 DS X | 1ST FLAG BYTE |
| 000001 | DS 3X | RESERVED |
| 000004 | PICPLEN DS H | LENGTH OF PICTURE |
| 000006 | PICHDR1 EQU *-PIC | |
| 000006 | PICTURE DS CL32 | PICTURE MAX LEN OF 32 |
| 00024 | PICLEN EQU ((*-PIC)/4)*4 | LENGTH OF PIC DSECT *LCB84097* |
| 0000A | PICLENF EQU ((*-PIC+3)/4) | # OF FULLWORDS *LCB84097* |

1.98 #PLFDS

```

COPY #PLFDS
*****
***          RELOCATABLE PAGE LIST FRAGMENT      ***
*****
Offset Value

000000    PLF      DSECT           12:58:15 04/04/85
000000    PLFID    DS   CL4        C'PLF*' 
000004    PLFNPLFA DS   A        ADDR OF NEXT PAGE LIST FRAGMENT
000008    PLFNPLSA DS   A        ADDR OF FIRST IN FREE SLOT LIST
00000C    PLFHIPLS DS   H        HIGHEST ACTIVE PAGE IN SLOT
00000E    PLFNMBER DS   H        PAGE LIST FRAGEMENT NUMBER
00010     PLFLEN   EQU   *-PLF  SIZE OF PAGE LIST FRAG PREFIX
00010     PLFFLEN  EQU   ((PLFLEN+3)/4)*4  SIZE OF PLF IN WORDS
00010     PLFPLS   EQU   *        START OF FIRST PAGE LIST SLOT
*****
***          PAGE LIST SLOT - REPRESENTS ONE STORAGE BLOCK      ***
***          ------      ***
***          WARNING: SEE MODULE ADSOARSM BEFORE CHANGING THE SIZE OF      ***
***          THIS CONTROL BLOCK.      ***
*****
000000    PLS      DSECT
000000    PLSLOCAD DS   A        FOR DC TO RETURN ADDR OF ADDR
000004    PLSPAGEA DS   0A       PAGE ADDR IF NOT USING RELO STG
000004    PLSNFREE DS   A        NEXT ON FREE PLS LIST
000008    PLSPAGNO DS   H        PAGE NUMBER OF THIS SLOT
00000A    PLSFRELN DS   H        FREE SPACE ON PAGE IN BYTES
00000C    PLSHLXEO DS   H        OFFSET TO END OF LINE INDEX
00000E    PLSFREEI DS   0XL1     SET IF SLOT FREE
000080    PLSFREEM EQU   X'80'
00000E    PLSFFLAGS DS   0X       FLAG BYTE
00000E    PLSFLXEO DS   H        OFFSET TO FIRST FREE LINE INDEX
00010     PLSLEN   EQU   *-PLS  SIZE OF ONE PAGE LIST SLOT
*****
***          LINE INDEX ENTRY      ***
*****
000000    LXE      DSECT
000000    LXEFREE  #FLAG X'80'  FREE LINE INDEX ENTRY
000000    LXEFREEI DS   0XL1
000080    LXEFREEM EQU   X'80'
000000    LXEFLAG  DS   0X
000000    LXEFFST DS   H        OFFSET TO START OF STG
000002    LXENLXEO DS   0H       OFFSET TO NEXT FREE LXE
000002    LXESTLEN DS   H        LENGTH OF STG
000004    LXELEN   EQU   *-LXE  LENGTH OF LINE INDEX ENTRY

```

1.99 #PMEDS

```

COPY #PMEDS
*****
*** PME IS A DSECT DEFINING THE HEADER FOR THE COMMAND TABLE ***
*** THAT COMPRISSES THE PREMAP PROCESS OF AN ADSO DIALOG. ***
*** IT EXISTS IN THE FIXED DIALOG BLOCK (FDB) FOR THE DIALOG ***
*** AND IS IMMEDIATELY FOLLOWED BY THE COMMAND TABLE. ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|-------------------------------------|
| 000000 | PME DSECT | 15:32:29 05/30/84 |
| 000000 | PMENAME DS CL32 | NAME OF THE PREMAP PROCESS |
| 000020 | PMELASTB DS F | OFFSET OF LAST BYTE IN PME |
| 000024 | PMERATA DS F | OFFSET TO READY AREA TABLE |
| 000028 | PMEFMCMEA DS F | OFFSET TO FIRST CME |
| 00002C | PMEPVER DS H | PROCESS VERSION |
| 00002E | PMENCMES DS H | NUMBER OF CME'S IN PREMAP PROCESS |
| 000030 | PMENEWF DS X | INITIALIZED TO X'FF' IF NEW FORMAT |
| | PRINT NOGEN | |
| | PMERDY #FLAG X'80' | AREAS NEED TO BE READIED |
| | PMERALL #FLAG X'40' | RAT HAS "READY ALL" BYTE *JEB84151* |
| 000031 | PMEEFLAG1 DS X | |
| | PRINT GEN | |
| 000032 | PMENMDTE DS H | NUMBER OF MODULE TABLE ENTRIES |
| 000034 | PMELNTA DS F | OFFSET TO CME LINE NUMBER TABLE |
| 000038 | PMEDATLU DS CL8 | DATE MODULE-067 LAST UPDATED |
| 000040 | PMEDATCR DS CL8 | DATE MODULE-067 CREATED |
| 000048 | PMEMDTA DS F | OFFSET TO INCLUDED MODULE TABLE |
| 00004C | PMEOFTBL DS F | OFFSET OF CME & EXEC CD OFFSET TBL |
| 000050 | PMEEEXTA DS F | RESERVED -ADDR FOR PME EXTENSION |
| 00054 | PMELEN EQU *--PME | LENGTH OF PME FIXED PORTION |

1.100 #PRBDS

```

COPY #PRBDS
*****
***          PRB: PRINT BUFFER FOR ADS/ONLINE
***
***      PRB IS A DSECT THAT DESCRIBES THE PRINT BUFFER USED BY
***      ADS/ONLINE AS A STAGING BUFFER FOR PROCESS SOURCE CARD
***      IMAGES. THE BUFFER IS BUILT DURING PROCESS PARSING AND
***      CONTAINS 5 ENTRY TYPES. THEY ARE: 1) SOURCE CARD IMAGE,
***      2) $ LINE DESCRIPTOR, I.E., A LIST OF UP TO 10 COLUMN
***      NUMBERS THAT POINT TO PARSE ERRORS, 3) A LIST OF
***      MESSAGE CODES, ONE FOR EVERY COLUMN NUMBER LISTED IN THE
***      ACCOMPANYING $ LINE DESCRIPTOR, 4) LINE SPACE COUNT, AND
***      5) PAGE EJECT.
***      FOR THE ONLINE GENERATOR, WHENEVER THE PRINT
***      BUFFER BECOMES FULL, IT IS WRITTEN AS A SCRATCH RECORD
***      AND THE SAME BUFFER IS USED TO HOLD SUBSEQUENT ENTRIES.
***      FOR THE BATCH GENERATOR, WHENEVER THE PRINT
***      BUFFER BECOMES FULL, A NEW PRINT BUFFER IS ACQUIRED AND
***      CHAINED TO THE PRECEDING PRINT BUFFER.
***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|---------------------|--------------------------------------|
| 000000 | PRB DSECT | 12/13/88 17:32:09 |
| 000000 | PRBNEXTA DS A | ADDR OF NEXT PRINT BUFF (BATCH) |
| 000004 | PRBNUM DS H | # OF ENTRIES IN PRINT BUFF |
| 000006 | PRBNABA DS H | NEXT AVAILABLE BYTE IN BUFF |
| 000008 | PRBFRSZ DS H | # OF UNUSED BYTES IN BUFF |
| 00000A | PRBHDRSZ EQU *-PRB | SIZE OF PRINT BUFF HEADER |
| 00000A | PRBSTRT DS 0X | START OF PRINT BUFF ENTRIES |
| 00FA0 | PRBUFFSZ EQU 4000 | SIZE OF ONLINE PRINT BUFFE*MCM87170* |
| 00FF0 | PRBTCHSZ EQU 4080 | SIZE OF BATCH PRINT BUFFER*GWG88336* |

1.101 #PSADS

```

COPY #PSADS
*****
***          PSA: PARSE STORAGE AREA (ADS/ONLINE)      ***
***          ***
***          PSA IS A DSECT THAT DESCRIBES THE WORK STORAGE AREAS USED   ***
***          BY THE PARSING MODULES IN THE ADS/ONLINE GENERATOR.        ***
***          ***
***          THE FIRST BLOCK OF VARIABLES IS REQUIRED BY THE           ***
***          GENERALIZED CONDITION AND EXPRESSION PARSE ROUTINES,     ***
***          ADSOGCND AND ADSOGEPR. IT MUST BE INCLUDED IN ANY          ***
***          OTHER PROGRAM WHICH MAKES USE OF THESE PARSE ROUTINES.    ***
***          ***
***          THE SECOND BLOCK OF VARIABLES MAY BE OPTIONALLY USED BY    ***
***          ANOTHER PROGRAM CALLING ADSOGCND AND ADSOGEPR. IT MAY BE    ***
***          USED TO HOLD THE XDE STACK HEADER AND XDE STACK ITSELF.    ***
***          ***
***          THE THIRD BLOCK OF VARIABLES IS REQUIRED ONLY FOR          ***
***          THE ADS/ONLINE GENERATOR.                                     ***
***          ***
***          THE DSECT HAS BEEN MODIFIED TO ACCCOMM07/06/90 16:32:06  04/01/94
***          PRODUCTS                                              ***
***          ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--|
| 000000 | PSA | DSECT |
| | * | 17:39:30 04/13/87 |
| | * | PSA BLOCK 1 |
| | * | |
| | * | VARIABLES REQUIRED BY ADSOGCND AND ADSOGEPR: |
| | * | |
| | * | THE FOLLOWING VARIABLES MUST BE INITIALIZED PRIOR TO CALLING |
| | * | ADSOGCND OR ADSOGEPR: |
| | * | PSASTKEA, PSAGGRDA, PSALEXLA, PSALTADA, PSAPSER, |
| | * | PSANXDEA, PSAXDEHA, PSAXDEEA, PSAIRATO, PSALITTO, |
| | * | PSANIRAO |
| | * | PSANDIMA (IF CONDITIONAL OPTIMIZER BEING CALLED) *MCM86122* |
| | * | |
| | * | THE FOLLOWING VARIABLES MUST BE CLEARED BEFORE THE CALL: |
| | * | PSALXDEA |
| | * | |
| | * | THE FOLLOWING VARIABLES ARE INTERFACE VARIABLES WITH THE |
| | * | LEXICAL ANALYZER: |
| | * | NXTTOKE, TOKEBUF, PACKED, PSANODEC, PSANODIG |
| | * | |
| | * | THE FOLLOWING VARIABLES ARE WORK VARIABLES: |
| | * | PHALF, PDBL, PSAOPN1L, PSAOPN1D, PSAOPN2L, PSAOPN2D |
| | * | PSACXDEA, PSACGCEA, PSASCSCV, PSAGRCP, PSACRNG, PSAFRST |
| | * | |
| | * | |
| 000000 | PSAID | DS CL4 |
| 000008 | PDBL | DS D |
| 000010 | PSASCSCV | DS F |
| 000014 | PSASTKEA | DS A |
| 000018 | PSAGGRDA | DS A |
| 00001C | PSALEXLA | DS A |
| 000020 | PSALTADA | DS A |
| 000024 | PSAPSER | DS A |
| 000028 | PSANXDEA | DS A |
| 00002C | PSALXDEA | DS A |
| 000030 | PSACXDEA | DS A |
| | | 'PSA*' |
| | | GENERALLY USEFUL DOUBLE WORD |
| | | SAVE AREA FOR STACK CHECKING |
| | | STACK END ADDRESS |
| | | ADDR OF GET GRD/GRE ROUTINE |
| | | ADDR OF LEXICAL ANALYSIS ROUTINE |
| | | ADDR OF LITERAL ADD ROUTINE |
| | | ADDR OF PARSE ERROR ROUTINE |
| | | ADDR FOR NEXT XDE TO BE BUILT |
| | | ADDR OF TOP XDE ON STACK |
| | | ADDR OF CONDITION XDE START |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|--------------------------------------|
| 000034 | PSAXDEHA DS A | ADDR OF XDE HDR FOR CURRENT STACK |
| 000038 | PSAXDEEA DS A | END ADDR OF XDE STACK |
| 00003C | PSACGCEA DS A | ADDR OF GCE FOR CURRENT CONDITION |
| 000040 | PSAENVBA DS A | ENVIRONMENT SAVE BUFFER ADDR |
| 000044 | PSACENVA DS A | CURRENT ENVIRONMENT SAVE ADDR |
| 000048 | PSAENVEA DS A | END ADDR OF ENVIRONMENT SAVE BUFF |
| 00004C | PSAESAVA DS A | ADDR ENVIRONMENT SAVE ROUTINE |
| 000050 | PSAERESA DS A | ADDR ENVIRONMENT RESTORE ROUTINE |
| 000054 | PSASAVSZ DS H | SIZE OF ENVIRONMENT SAVE AREA |
| 000056 | PHALF DS H | GENERALLY USEFUL HALFWORD |
| 000058 | PSAIRATO DS H | REC TABLE OFFSET TO IRA ADDRESS |
| 00005A | PSALITTO DS H | REC TABLE OFFSET TO LITPOOL ADDR |
| 00005C | PSANIRAO DS H | NEXT AVAILABLE OFFSET IN IRA |
| 00005E | PSAOPN1L DS H | TOTAL # DIGITS IN 1ST OPERAND |
| 000060 | PSAOPN1D DS H | # OF DECIMAL PLACES IN 1ST OPND |
| 000062 | PSAOPN2L DS H | TOTAL # DIGITS IN 2NS OPERAND |
| 000064 | PSAOPN2D DS H | # OF DECIMAL PLACES IN 2ND OPND |
| 000066 | NXTTOKE DS H | NEXT AVAILABLE SLOT FOR TOKEN |
| 000068 | TOKEBUF DS 2CL256 | TOKEN BUFFER |
| 00200 | TOKEBLEN EQU *-TOKEBUF | LENGTH OF TOKEN BUFFER |
| 000268 | PACKED DS XL9 | HOLDING BUFFER FOR PACKED DEC # |
| 000271 | PSANODEC DS X | # DECIMAL PLACES |
| 000272 | PSANODIG DS X | # OF DIGITS IN DEC # |
| | PSAGRPC #FLAG X'80' | ON IF SUBS ELEM IS OCC GROUP |
| 000273 | PSAGRPCI DS 0XL1 | |
| | 00080 PSAGRPCM EQU X'80' | |
| | PSACRNG #FLAG X'40' | ON IF CONDITION IS FOR RANGE |
| 000273 | PSACRNGI DS 0XL1 | |
| | 00040 PSACRNGM EQU X'40' | |
| | PSAFRST #FLAG X'20' | ON IF 1ST HALF OF RANGE COND |
| 000273 | PSAFRSTI DS 0XL1 | |
| | 00020 PSAFRSTM EQU X'20' | |
| | PSAMCND #FLAG X'10' | ON IF MULTIPLE VALUES FOR A COND |
| 000273 | PSAMCNDI DS 0XL1 | |
| | 00010 PSAMCNDM EQU X'10' | |
| | * USE OF PSAMBB FLAG IN GEXP REPLACED WITH AN EXPLICIT | *MCM86176* |
| | * CHECK OF XDEDATYP FOR MULTI-BIT BINARY | *MCM86176* |
| | * PSAMBB #FLAG X'08' | FIELD IS MULTIBIT BINARY *MCM86176* |
| | PSAMBBE #FLAG X'08' | MULTIBIT BINARY ERROR *ADS370/506 |
| 000273 | PSAMBBEI DS 0XL1 | |
| | 00008 PSAMBBEM EQU X'08' | |
| | PSAPRFX #FLAG X'04' | ON IF PREFIX VARIABLE *MCM86181* |
| 000273 | PSAPRFXI DS 0XL1 | |
| | 00004 PSAPRFXM EQU X'04' | |
| | * | BEING HANDLED IN ADSOGMSG*MCM86181* |
| | PSASUBF #FLAG X'02' | ON IF processing subscript MCM87 |
| 000273 | PSASUBFI DS 0XL1 | |
| | 00002 PSASUBFM EQU X'02' | |
| 000273 | PSAEXPF DS X | FLAG BYTE FOR ADSOGEXP |
| 000274 | DS 0F | |
| 000274 | PSACGVEA DS A | ADDR OF GVE FOR CURRENT COND VALUE |
| 000278 | PSACGRDA DS A | ADDR OF GRD FOR CURRENT COND VALUE |
| 00027C | PSAMSWTO DS H | REC TBL OFFSET TO MAP SWITCH REC |
| | PSA1SUB #FLAG X'80' | 1ST UNRESOLVED SUBRTN FND *MET88112* |
| 00027E | PSA1SUBI DS 0XL1 | |
| | 00080 PSA1SUBM EQU X'80' | |
| | LEXFLTS #FLAG X'20' | SHORT FLOAT CONSTANT PARSE*LCB84229* |
| 00027E | LEXFLTSI DS 0XL1 | |
| | 00020 LEXFLTSM EQU X'20' | |
| | LEXFLTL #FLAG X'10' | LONG FLOAT CONSTANT PARSED*LCB84229* |
| 00027E | LEXFLTLI DS 0XL1 | |
| | 00010 LEXFLTLM EQU X'10' | |
| | LEXFLAG DS X | FLAG BYTE FOR FLOAT CONST *ISS86527* |
| | PSASQLC #FLAG X'80' | ON IF SQL CMND PARSING **SQL** |
| 00027F | PSASQLCI DS 0XL1 | |
| | 00080 PSASQLCM EQU X'80' | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|---------------------------------------|
| | | GNLX(GETCARD) INPUT FRM MEMRY**SQL** |
| 00027F | PSASQIN #FLAG X'40' | |
| | PSASQINI DS 0XL1 | |
| 00040 | PSASQINM EQU X'40' | |
| | PSASQIQ #FLAG X'20' | ON IF IN SQL QUOTED STRING **SQL** |
| 00027F | PSASQIQI DS 0XL1 | |
| 00020 | PSASQIQM EQU X'20' | |
| | PSASQEX #FLAG X'10' | STRING 'EXEC SQL' CLEARED *JMA91256* |
| 00027F | PSASQEXI DS 0XL1 | |
| 00010 | PSASQEXM EQU X'10' | |
| | PSASQI2 #FLAG X'08' | ON IF QUOTED STRING FOUND ADS370/199 |
| 00027F | PSASQI2I DS 0XL1 | |
| 00008 | PSASQI2M EQU X'08' | |
| 00027F | PSASQLF DS X | SQL COMMANDS FLAG *LMA87245* |
| 000280 | PSANDIMA DS A | ADDR OF GET NUM DIMS RTNE=MCM86122* |
| 000284 | PSAGEXPA DS F | ARITHM. EXPR. PARSER ADDR *LMA86309* |
| 000288 | PSACMXAD DS F | REPORTER COMMUNIC AREA ADDR*LMA86317* |
| 00028C | PSACNDAD DS F | CONDITION EXPR.PARSER ADDR*LMA86317* |
| 000290 | PSASQLOS DS A | SQL ORIGINAL SYNTAX **SQL** |
| 000294 | PSASQLPS DS A | SQL NEW PRE-COMPILED SYNTAX **SQL** |
| 000298 | PSASQLNO DS H | SQL NUMB LINES(ORIG SYNTAX) **SQL** |
| 00029A | PSASQLPN DS H | SQL NUMB LINES(PRECOMPILED) **SQL** |
| 0029C | PSALEN1 EQU *-PSA | FIRST PSA BLOCK SIZE |
| * | | PSA BLOCK 2 |
| * | | |
| * | | |
| * | * THE FOLLOWING BUFFERS MAY BE USED TO BUILD THE XDE STACK HEADER | |
| * | * AND XDE STACK ITSELF. ADSOGCND AND ADSOGEXP DO NOT ADDRESS | |
| * | * THESE BUFFERS BY NAME SO THE USER MAY USE A DIFFERENT AREA FOR | |
| * | * THE XDE STACK HEADER AND XDE STACK. THE ONLY REQUIREMENT IS THAT | |
| * | * PRIOR TO CALLING ADSOGCND AND ADSOGEXP THE FOLLOWING VARIABLES | |
| * | * HAVE BEEN INITIALIZED TO THE FOLLOWING VALUES: | |
| * | * PSAXDEHA - ADDRESS OF THE XDE STACK HEADER | |
| * | * PSANXDEA - ADDRESS OF THE XDE STACK START | |
| * | * AFTER CALLING ADSOGCND OR ADSOGEXP THE XDE STACK HEADER HAS | |
| * | * THE FOLLOWING LAYOUT: | |
| * | *----- | |
| * | 1 # OF XDE'S IN STACK | 1 # IRA BYTES REQUIRED 1 |
| * | ----- | |
| * | 1 # BYTES IN XDE STACK | 1 UNUSED 1 |
| * | ----- | |
| * | | |
| 00029C | PSABLK2 DS 0F | |
| 00029C | PSAVAR1H DS 2F | VAR-1 XDE STACK HEADER |
| 0002A4 | PSAVAR1N DS 300F | VAR-1 XDE STACK *MCM86163* |
| 004B0 | PSAVAR1S EQU *-PSAVAR1N | SIZE OF VAR-1 XDE STACK |
| 00754 | PSALEN2 EQU *-PSA | SECOND PSA BLOCK SIZE |
| * | PSA BLOCK 3 | |
| * | | |
| * | | |
| * | * THE FOLLOWING VARIABLES ARE APPLICABLE ONLY TO THE ADS/ONLINE | |
| * | * GENERATOR PARSE MODULES. | |
| * | | |
| 000754 | PSABLK3 DS 0F | |
| 000754 | PSAVAR2H DS 2F | VAR-2 XDE STACK HEADER |
| 00075C | PSAVAR2N DS 301F | VAR-2 XDE STACK *MCM86163* |
| 004B4 | PSAVAR2S EQU *-PSAVAR2N | SIZE OF VAR-2 XDE STACK |
| 000C10 | PSAPLIST DS 10F | DC MACRO PARM AREA |
| 000C38 | PSADLGNM DS CL8 | CONTROL CMD DLG/PGM NAME (LITERAL) |
| 000C40 | PSAWPNUM DS F | SAVE AREA FOR NUMBER |
| 000C44 | PSASAVRG DS F | TEMP REGISTER STORAGE AREA |
| 000C48 | PSAFSXDA DS F | 1ST SUBSCRIPT XDE STK ADDR |
| 000C4C | PSASUBV DS F | SUBSCRIPT EVALUATION ACCUM |
| 000C50 | PSASUBM DS F | SUBS DIMENSION MULTIPLIER |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|---|
| 000C54 | PSAFSEQN DS | F |
| 000C58 | PSATXTKY DS | F |
| 000C5C | PSASCRID DS | F |
| 000C60 | PSASTNUM DS | F |
| 000C64 | PSAAXDEA DS | A |
| 000C68 | PSAERCLA DS | A |
| 000C6C | PSANXTPA DS | A |
| 000C70 | PSACCNTA DS | A |
| 000C74 | PSAWPRET DS | A |
| 000C78 | PSAPATCH DS | A |
| 000C7C | PSAGMFDA DS | A |
| 000C80 | PSAPBFMA DS | A |
| 000C84 | PSACPROA DS | A |
| 000C88 | PSAGSAA DS | A |
| 000C8C | PSAGWKA DS | A |
| 000C90 | PSAGREA DS | A |
| 000C94 | PSAFWAEA DS | A |
| 000C98 | PSACGCMa DS | A |
| 000C9C | PSACCMEA DS | A |
| 000CA0 | PSANPMLA DS | A |
| 000CA4 | PSANPMEA DS | A |
| 000CA8 | PSACSDEA DS | A |
| 000CAC | PSANSXDA DS | A |
| 000CB0 | PSAPRTBA DS | A |
| 000CB4 | LEXADDR DS | A |
| 000CB8 | PSALGLEA DS | A |
| 000CBC | PSAFWAA DS | A |
| 000CC0 | ORG | PSAFWAA |
| 000CBC | PSACPRA DS | A |
| 000CC0 | PSAFFTA DS | A |
| 000CC4 | PSAGRDA DS | A |
| 000CC8 | PSAGSSA DS | A |
| 000CCC | PSAADEA DS | A |
| 000CD0 | PSACMFA DS | A |
| 000CD4 | PSAKWLIT DS | H |
| 000CD6 | PSADATLN DS | H |
| 000CD8 | PSAVDBTO DS | H |
| 000CDA | PSAVRETO DS | H |
| 000CDC | PSAOBTBTO DS | H |
| 000CDE | COMPERRS DS | H |
| 000CE0 | NXTNUMB DS | H |
| 000CE2 | LEXLNTH DS | H'0' |
| 000CE4 | LEXCOLM DS | H'72' |
| 000CE6 | FSTCOLM DS | H'01' |
| 000CE8 | LSTCOLM DS | H'72' |
| 000CEA | PSACMESZ DS | H |
| 000CEC | PSAPMESZ DS | H |
| 000CEE | PSAOCCNO DS | H |
| 000CF0 | PSANIF DS | H |
| 000CF2 | PSANWHLE DS | H |
| 000CF4 | PSANEXIT DS | H |
| 000CF6 | PSANNEXT DS | H |
| 000CF8 | PSARATIX DS | H |
| 000CFA | PSAIRALN DS | H |
| 000CFc | PSADTABO DS | H |
| 000D00 | NUMBER DS | 10F |
| | * DLRBUFF, MSGCBUFF, AND SRCBUFF ARE BUILD AREAS FOR ENTRIES | |
| | * INTO THE PRINT BUFFER FOR COMPILED PROCESS SOURCE | |
| 000D28 | DLRBUFF DS | OF |
| | DLRFLG #FLAG X'40' | BUILD AREA FOR \$ LINE ENTRY INDICATES \$ LINE ENTRY |
| 000D28 | DLRFLGI DS | 0XL1 |
| 00040 | DLRFLGM EQU | X'40' |
| 000D28 | DLRFLGB DS | X |
| 000D29 | DLRNUM DS | X |
| 00002 | DLRHDRSZ EQU | *-DLRBUFF |
| 000D2A | DLRCOL DS | 10X |
| | DOLLAR FLAG BYTE | |
| | NUMBER OF \$'S IN ENTRY | |
| | SIZE OF \$ ENTRY HEADER | |
| | COLUMN #'S FOR \$'S | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|---|
| 000D34 | 0000A DLRMAX EQU *-DLRCOL MSGCBUFF DS OF MSGCFLG #FLAG X'20' | MAX # OF ENTRIES IN DLRCBUFF BUILD AREA FOR MSG CODE ENTRY INDICATES MSG CODE ENTRY |
| 000D34 | 00020 MSGCFLGM EQU X'20' MSGCFLGB DS X MSGCNUM DS X | MESSAGE CODE FLAG BYTE NUMBER OF MSG CODES IN ENTRY |
| 000D35 | 00002 MSGHDRSZ EQU *-MSGCBUFF MSGCODES DS 40X | |
| 000D60 | SRCBUFF DS OF SRCFLG #FLAG X'80' | BUILD AREA FOR SOURCE ENTRY INDICATES SOURCE ENTRY |
| 000D60 | 00080 SRCFLGM EQU X'80' SRCFLGB DS X | MSG CODE NUMBERS SOURCE FLAG BYTE |
| 000D61 | SRCSEQNO DS 7X FRSTCRD #FLAG X'80' | SOURCE SEQUENCE NUMBER ON IF NO PROCESS CARD READ YET |
| 000D68 | 00080 FRSTCRDI DS 0XL1 SRCERR #FLAG X'40' | |
| 000D68 | 00040 SRCERRI DS 0XL1 SRCERRM EQU X'40' | ON IF CURRENT CARD HAS ERROR |
| 000D68 | 00020 PRINTEDI DS 0XL1 MXLERR #FLAG X'10' | ON IF CARD IMAGE NOT TO BE PRNTED |
| 000D68 | 00010 MXLERRI DS 0XL1 MXLERRM EQU X'10' | ON IF MAX # ERRS FOR LINE FOUND |
| 000D68 | 00004 PSATGT #FLAG X'04' | ON IF PROCESSING TARGET FL*LCB84096* |
| 000D68 | 00004 PSATGTM EQU X'04' | |
| 000D68 | 00002 PSAETGT #FLAG X'02' | ON IF PROCESSING EDIT TRGT*LCB84096* |
| 000D68 | 00002 PSAETGTM EQU X'02' | |
| 000D68 | 00001 PSAPTGT #FLAG X'01' | ON IF FINISHED PROC TARGET*LCB84096* |
| 000D68 | 00001 PSAPTGTM EQU X'01' | |
| 000D68 | PSAPRBFL DS X | PRINT BUFF FLAG BYTE |
| 000D69 | PSADATYP DS X | IRA DATA TYPE |
| 000D6A | DELIM DS C'*' | COMMENT DELIMITER CHARACTER |
| 000D6B | HOLDCC DS CL1 | |
| 000D6C | ZONEDNO DS XL16 | HOLDING BUFFER FOR ZONED DEC # |
| 000D7C | PSAOFCOD DS X | OP CODE FOR PROCEDURAL CME |
| | PSADXB #FLAG X'80' | ON IF CURRENT XDE STK HAS DXB |
| 000D7D | 00080 PSADXBI DS 0XL1 PSADXB #FLAG X'80' | |
| 000D7D | 00040 PSAV1DX #FLAG X'40' | ON IF VAR-1 XDE STK HAS A DXB |
| 000D7D | 00040 PSAV1DXI DS 0XL1 | |
| 000D7D | 00020 PSAV1DXM EQU X'40' | |
| 000D7D | PSAV2DX #FLAG X'20' | ON IF VAR-2 XDE STK HAS A DXB |
| 000D7D | 00020 PSAV2DXI DS 0XL1 | |
| 000D7D | 00020 PSAV2DXM EQU X'20' | |
| 000D7D | PSASGFD #FLAG X'10' | ON IF TEST FOR SINGLE MAPFIELD |
| 000D7D | 00010 PSASGFDM EQU X'10' | |
| 000D7D | PSAV3DX #FLAG X'08' | ON IF VAR-3 XDE STK HAS A DXB |
| 000D7D | PSAV3DXI DS 0XL1 | |
| 000D7D | 00008 PSAV3DXM EQU X'08' | |
| 000D7D | PSAV4DX #FLAG X'04' | ON IF VAR-4 XDE STK HAS A DXB |
| 000D7D | PSAV4DXI DS 0XL1 | |
| 000D7D | 00004 PSAV4DXM EQU X'04' | |
| 000D7D | PSAXDXB #FLAG X'02' | ON IF PARSE OF MFLD IS BANNED |
| 000D7D | PSAXDXBI DS 0XL1 | |
| 000D7D | 00002 PSAXDXBM EQU X'02' | |
| 000D7D | PSAPMIL #FLAG X'01' | ON IF MAPFIELD LIST BEING BUILT |
| 000D7D | PSAPMILI DS 0XL1 | |
| | 00001 PSAPMILM EQU X'01' | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------------------|--|
| 000D7D | PSADXBLF DS X PSADEF #FLAG X'80' | FLAG BYTE FOR DXB CONSTRUCTION DEFINE STMT HAS BEEN PROCESSED |
| 000D7E | PSADEFI DS 0XL1 | |
| 000080 | PSADEFM EQU X'80' | |
| | PSA2DEF #FLAG X'40' | DUPPLICATE SUBROUTINE DEFINITION |
| 000D7E | PSA2DEFI DS 0XL1 | |
| 000040 | PSA2DEFM EQU X'40' | |
| | PSACALL #FLAG X'20' | SUBROUTINE REFERENCED BY CALL |
| 000D7E | PSACALLI DS 0XL1 | |
| 000020 | PSACALLM EQU X'20' | |
| | PSAIMPE #FLAG X'10' | IMPLICIT 'END' FOR THEN/ELSE |
| 000D7E | PSAIMPEI DS 0XL1 | |
| 000010 | PSAIMPEM EQU X'10' | |
| | PSANXTP #FLAG X'08' | ON IF PARM BUFFER HAS NEXT ADDRS |
| 000D7E | PSANXTP1 DS 0XL1 | |
| 000008 | PSANXTPM EQU X'08' | |
| | PSAEXIP #FLAG X'04' | ON IF PARM BUFFER HAS EXIT ADDRS |
| 000D7E | PSAEXIPI DS 0XL1 | |
| 000004 | PSAEXIPM EQU X'04' | |
| | PSAEXIT #FLAG X'02' | ON IF PASSING EXIT ADDRS TO IF |
| 000D7E | PSAEXITI DS 0XL1 | |
| 000002 | PSAEXITM EQU X'02' | |
| 000D7E | PSAFLOFL DS X PSASTOL #FLAG X'80' | ADSOGFLO FLAG BYTE DB STORE PRIVACY LOCK |
| 000D7F | PSASTOLI DS 0XL1 | |
| 000080 | PSASTOLM EQU X'80' | |
| | PSAKEPL #FLAG X'40' | DB KEEP PRIVACY LOCK |
| 000D7F | PSAKEPLI DS 0XL1 | |
| 000040 | PSAKEPLM EQU X'40' | |
| | PSADISL #FLAG X'20' | DB DISCONNECT PRIVACY LOCK |
| 000D7F | PSADISLI DS 0XL1 | |
| 000020 | PSADISLM EQU X'20' | |
| | PSAMODL #FLAG X'10' | DB MODIFY PRIVACY LOCK |
| 000D7F | PSAMODLI DS 0XL1 | |
| 000010 | PSAMODLM EQU X'10' | |
| | PSACONL #FLAG X'08' | DB CONNECT PRIVACY LOCK |
| 000D7F | PSACONLI DS 0XL1 | |
| 000008 | PSACONLM EQU X'08' | |
| | PSAGETL #FLAG X'04' | DB GET PRIVACY LOCK |
| 000D7F | PSAGETLI DS 0XL1 | |
| 000004 | PSAGETLM EQU X'04' | |
| | PSAFNDL #FLAG X'02' | DB FIND PRIVACY LOCK |
| 000D7F | PSAFNDLI DS 0XL1 | |
| 000002 | PSAFNDLM EQU X'02' | |
| | PSAERAL #FLAG X'01' | DB ERASE PRIVACY LOCK |
| 000D7F | PSAERALI DS 0XL1 | |
| 000001 | PSAERALM EQU X'01' | |
| 000D7F | PSAPLFLG DS X PSARDYR #FLAG 0 | DB PRIVACY LOCK FLAG BYTE READY RETRIEVAL |
| 000D80 | PSARDYRI DS 0XL1 | |
| 000000 | PSARDYRM EQU 0 | |
| | PSARDYU #FLAG 1 | READY UPDATE |
| 000D80 | PSARDYUI DS 0XL1 | |
| 000001 | PSARDYUM EQU 1 | |
| | PSARDYP #FLAG 2 | READY PROTECTED |
| 000D80 | PSARDYPI DS 0XL1 | |
| 000002 | PSARDYPM EQU 2 | |
| | PSARDYE #FLAG 4 | READY EXCLUSIVE |
| 000D80 | PSARDYEI DS 0XL1 | |
| 000004 | PSARDYEM EQU 4 | |
| | PSARDNO #FLAG 6 | READY NOREADY *NOREADY |
| 000D80 | PSARDNOI DS 0XL1 | |
| 000006 | PSARDNOM EQU 6 | |
| 000D80 | PSARDYPL DS X PSAFSNF #FLAG X'80' | READY PRIVACY LOCK INDEX FIND <SEQUENCE-NUMBER> |
| 000D81 | PSAFSNFI DS 0XL1 | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------|------------------------------------|
| | 00080 PSAFSNFM EQU X'80' | |
| | PSAFSVF #FLAG X'40' | FIND <SEQUENCE-VARIABLE> |
| 000D81 | PSAFSVFI DS OXL1 | |
| | 00040 PSAFSVFM EQU X'40' | |
| | PSAFDBK #FLAG X'20' | FIND DB-KEY |
| 000D81 | PSAFDBKI DS OXL1 | |
| | 00020 PSAFDBKM EQU X'20' | |
| | PSAFSK #FLAG X'10' | FIND USING <SORT-KEY> |
| 000D81 | PSAFSKI DS OXL1 | |
| | 00010 PSAFSKM EQU X'10' | |
| | PSAFRFL #FLAG X'08' | PARSE FOR FIND <RECORD> FAILED |
| 000D81 | PSAFRFLI DS OXL1 | |
| | 00008 PSAFRFLM EQU X'08' | |
| | PSANSEQ #FLAG X'04' | NEGATIVE SEQUENCE NUMBER EXPECTED |
| 000D81 | PSANSEQI DS OXL1 | |
| | 00004 PSANSEQM EQU X'04' | |
| | PSAKEPX #FLAG X'02' | KEEP EXCLUSIVE COMMAND |
| 000D81 | PSAKEPXi DS OXL1 | |
| | 00002 PSAKEPXM EQU X'02' | |
| | PSAITOV #FLAG X'01' | ASSIGN IRA TO VARIABLE |
| 000D81 | PSAITOVI DS OXL1 | |
| | 00001 PSAITOVm EQU X'01' | |
| 000D81 | PSADABFL DS X | ADSOGDAB FLAG BYTE |
| | PSAMFOR #FLAG X'80' | MODIFY MAP FOR CLAUSE FLAG |
| 000D82 | PSAMFORI DS OXL1 | |
| | 00080 PSAMFORM EQU X'80' | |
| | PSAMEXC #FLAG X'40' | MODIFY MAP EXCEPT FLAG |
| 000D82 | PSAMEXCi DS OXL1 | |
| | 00040 PSAMEXCM EQU X'40' | |
| | PSAMATT #FLAG X'20' | MODIFY MAP ATTRIBUTES CLAUSE FLAG |
| 000D82 | PSAMATTi DS OXL1 | |
| | 00020 PSAMATTM EQU X'20' | |
| | PSAEXTM #FLAG X'10' | ON IF EXTENDED MRB/MRE REQUIRED |
| 000D82 | PSAEXTMI DS OXL1 | |
| | 00010 PSAEXTMM EQU X'10' | |
| 000D82 | PSAMAPFL DS X | ADSOGMAP FLAG BYTE |
| | FAIL #FLAG X'80' | ON IF TOKEN FAILS TEST |
| 000D83 | FAILi DS OXL1 | |
| | 00080 FAILM EQU X'80' | |
| | PSARERR #FLAG X'40' | ON IF ERROR MSGS REMAIN IN MSG BUF |
| 000D83 | PSARERRi DS OXL1 | |
| | 00040 PSARERRM EQU X'40' | |
| | PSACMSP #FLAG X'20' | ON IF CME HAS SPACE AVAILABLE |
| 000D83 | PSACMSPi DS OXL1 | |
| | 00020 PSACMSPM EQU X'20' | |
| | EOF #FLAG X'10' | SOURCE INPUT COMPLETE |
| 000D83 | EOFi DS OXL1 | |
| | 00010 EOFM EQU X'10' | |
| | PSACCMD #FLAG X'08' | ON IF CONTROL CMD PARSED |
| 000D83 | PSACCMDi DS OXL1 | |
| | 00008 PSACCMDM EQU X'08' | |
| | PSAIVCM #FLAG X'04' | ON IF CURRENT CMD IS INVOKE |
| 000D83 | PSAIVCMI DS OXL1 | |
| | 00004 PSAIVCMM EQU X'04' | |
| | EOFOK #FLAG X'02' | ON IF EOF TO BE IGNORED HERE |
| 000D83 | EOFOKi DS OXL1 | |
| | 00002 EOFOKM EQU X'02' | |
| | PSACMPU #FLAG X'01' | ON IF COMPUTE COMMAND |
| 000D83 | PSACMPUI DS OXL1 | |
| | 00001 PSACMPUM EQU X'01' | |
| | PSAFLAG1 DS X | FIRST FLAG BYTE |
| | PSAXDES #FLAG X'80' | XDE STACK FOLLOWING FORMAT 3 ADE |
| 000D84 | PSAXDESI DS OXL1 | |
| | 00080 PSAXDESM EQU X'80' | |
| | PSAXPTR #FLAG X'40' | XDE STACK JUST POINTS TO A FIELD |
| 000D84 | PSAXPTRi DS OXL1 | |

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|--------------------------------------|
| 00040 | PSAXPTRM EQU X'40' | |
| | PSAGSQK #FLAG X'20' | GET SCRATCH/QUEUE KEEP OPTION |
| 000D84 | PSAGSQKI DS 0XL1 | |
| 00020 | PSAGSQKM EQU X'20' | |
| | PSAWPCL #FLAG X'10' | WRITE PRINTER CLASS SPECIFIED |
| 000D84 | PSAWPCLI DS 0XL1 | |
| 00010 | PSAWPCLM EQU X'10' | |
| | PSAINTR #FLAG X'08' | ON IF INIT REC LIST IN PROGESS |
| 000D84 | PSAINTRI DS 0XL1 | |
| 00008 | PSAINTRM EQU X'08' | |
| 000D84 | PSAMISFL DS X | ADSOGMIS FLAG BYTE |
| | PSALRWH #FLAG X'80' | INDICATE CONDITION IS FOR LR VERB |
| 000D85 | PSALRWHI DS 0XL1 | |
| 00080 | PSALRWHM EQU X'80' | |
| | PSADNLRL #FLAG X'40' | INDICATE DATANAME OF LR PARSED |
| 000D85 | PSADNLRI DS 0XL1 | |
| 00040 | PSADNLRM EQU X'40' | |
| | PSAOFLR #FLAG X'20' | INDICATE XDE STACK HAS OF LR DNM |
| 000D85 | PSAOFLRI DS 0XL1 | |
| 00020 | PSAOFLRM EQU X'20' | |
| | PSARDYA #FLAG X'10' | READY ALL HAS BEEN PARSED |
| 000D85 | PSARDYAI DS 0XL1 | |
| 00010 | PSARDYAM EQU X'10' | |
| 000D85 | PSALRFLG DS X | LOGICAL RECORD/DATA BASE FLAG BYTE |
| | PSAABTC #FLAG X'80' | ON IF ABORT CME UNDER CONSTRUCTION |
| 000D86 | PSAABTCI DS 0XL1 | |
| 00080 | PSAABTCM EQU X'80' | |
| | PSAABTM #FLAG X'40' | ON IF ABORT MESSAGE SPECIFIED |
| 000D86 | PSAABTMI DS 0XL1 | |
| 00040 | PSAABTMM EQU X'40' | |
| | PSAUFNC #FLAG X'20' | ON IF USER FUNC BEING PARS*LCB84138* |
| 000D86 | PSAUFNCF DS 0XL1 | |
| 00020 | PSAUFNCM EQU X'20' | |
| 000D86 | PSAABTFL DS X | ABORT FLAG |
| 000D88 | PSACINCB DS A | CURR CONTROL BLK W/ INCLUDE INFO |
| 000D8C | PSAFINCB DS A | 1ST CONTROL BLK W/ INCLUDE INFO |
| 00D90 | PSAOLEN EQU (*-PSA) | ONLINE PSA FIXED LENGTH PORTION*GWG* |
| | ***** FOLLOWING FIELDS ARE USED IN ADSOBGPS ***** | |
| 000D90 | PSASTRCD DS CL32 | STATUS RCD NAME READ USED ADSOBGPS |
| 000DB0 | PSASTRVR DS H | STATUS RCD VER READ USED ADSOBGPS |
| 000DB2 | PSASBFLG DS CL1 | FLAG FOR ADD/MOD/DEL IN ADSOBGPS |
| 000DB3 | PSAUTOST DS CL1 | AUTO STATUS READ IN,USED ADSOBGPS |
| 000DB4 | PSAMPNM DS CL8 | TEMP STORAGE FOR MAPNAME (ADSOBGP) |
| 000DBC | PSAMPVR DS H | TEMP STORAGE FOR MAPVER (ADSOBGP) |
| 000DBE | PSAMPFLG DS CL1 | FLAG FOR ADD/MOD/DEL FOR MAP(") |
| | PSAMPRC #FLAG X'80' | ON IF MAP PROCESSING HAS BEEN DONE |
| 000DBF | PSAMPRCI DS 0XL1 | |
| 00080 | PSAMPRCM EQU X'80' | |
| 000DBF | PSAFLAG DS X | FLAG FOR MAP PROCESSING |
| | ***** END OF FIELDS USED IN ADSOBGPS ***** | |
| 00DC0 | PSAO2 EQU * | ONLINE PSA SECOND FIXED PORTION*GWG* |
| 000DC0 | PSAXXDEA DS A | ADDR OF EXTRA XDE STACK AREA |
| 000DC4 | LEXLNTH2 DS H | LEXLNTH SAVE AREA FOR STR/BIN STR |
| | LEXSTR #FLAG X'80' | STRING CONSTANT ALREADY PARSED |
| 000DC6 | LEXSTRI DS 0XL1 | |
| 00080 | LEXSTRM EQU X'80' | |
| | LEXBIN #FLAG X'40' | BINARY STRING CONSTANT IS PARSED |
| 000DC6 | LEXBINI DS 0XL1 | |
| 00040 | LEXBINM EQU X'40' | |
| 000DC6 | LEXFLAG1 DS X | FLAG BYTE FOR STR/BIN CONSTANTS |
| 000DC7 | PSABINL DS X | LENGTH OF MULTIBIT BINARY FIELD |
| 000DC8 | PSABIN DS F | MULTIBIT BINARY VALUE |
| 000DCC | LEXCOLM2 DS H | LEXCOLM SAVE AREA FOR STR/BIN STR |
| 000DCE | PSAEMSKL DS H | LENGTH OF MASK WORK AREA *LCB84123* |
| 000DD0 | PSAEMSKS DS F | STORAGE FOR MASK WORK AREA*LCB84123* |
| 000DD4 | PSAPWRCT DS X | CT OF HIGHEST EXPR POWER *LCB84125* |

| <u>Offset</u> | <u>Value</u> | |
|---------------|---|---|
| 000DD5 | PSADECCT DS X | CT OF HIGHEST # OF DECIMAL *LCB84125* |
| 000DD6 | PSAFNDC DS X | FUNC RSLT VALUE # OF DEC *LCB84143* |
| 000DD7 | PSAFDATP DS X | FUNC RSLT VALUE DATA TYPE *LCB84143* |
| 000DD8 | PSAEVALS DS F | AMT OF SPACE NEEDED BY EVL *LCB84125* |
| 000DDC | PSASAVF1 DS F | SAVE WORD *LCB84125* |
| 000DE0 | PSACMXDE DS A | ADDR OF CURR MODEL XDE *LCB84139* |
| 000DE4 | PSAFDATL DS H | FUNC RSLT VALUE DATA LEN *LCB84143* |
| | PSAO PDL #FLAG X'80' | RSLT LEN = OPND *LCB84143* |
| 000DE6 | PSAO PDL1 DS OXL1 | |
| | 00080 PSAOPDLM EQU X'80' | |
| | PSAO PDC #FLAG X'40' | RSLT # OF DEC = OPND *LCB84143* |
| 000DE6 | PSAO PDC1 DS OXL1 | |
| | 00040 PSAOPDCM EQU X'40' | |
| | PSAO PDT #FLAG X'20' | RSLT DATA TYPE = OPND *LCB84143* |
| 000DE6 | PSAO PDTI DS OXL1 | |
| | 00020 PSAOPDTM EQU X'20' | |
| | PSACLCL #FLAG X'10' | RSLT LEN = CALC *LCB84143* |
| 000DE6 | PSACLCLI DS OXL1 | |
| | 00010 PSACLCLM EQU X'10' | |
| 000DE6 | PSAFFLG1 DS X | FUNC FLAG BYTE *LCB84143* |
| 000DE7 | PSAPARCT DS X | PAREN COUNTER DURING PARSE *MCM87020* |
| 000DE8 | DS F | RESERVED *LCB84236* |
| 000DEC | INTFLOAT DS 2F | INTERNAL FLOAT CONSTANT *LCB84230* |
| 000DF4 | DSPFLOAT DS 6F | DISP FLOAT CONSTANT WORKAR *LCB84236* |
| 000E0C | PSAEXDEA DS F | EXTRA XDE ADDR FOR COBOL M *LCB84123* |
| 000E10 | PSAPICA DS A | ADDR OF GRE TARGET PICTURE *LCB84097* |
| 000E14 | PSACGMDA DS F | CURRENT GMD POINTER |
| 000E18 | PSAMDNDX DS H | HIGHEST MODULE INDEX |
| 000E1A | PSACRNDX DS H | CURRENT MODULE INDEX |
| 000E1C | PSAPRNDX DS H | PREVIOUS MODULE INDEX |
| | PSAINON #FLAG X'80' | COMMAND IS AN INCLUDE |
| 000E1E | PSAINONI DS OXL1 | |
| | 00080 PSAINONM EQU X'80' | |
| | PSAINOF #FLAG X'40' | COMMAND IS LAST STMT IN INCLUDE |
| 000E1E | PSAINOFI DS OXL1 | |
| | 00040 PSAINOFM EQU X'40' | |
| 000E1E | DS X | FLAG FOR INCLUDE PROCESSING ON IF ROUNDED OPTION |
| | PSAROUN #FLAG X'80' | |
| 000E1F | PSAROUNI DS OXL1 | |
| | 00080 PSAROUNM EQU X'80' | |
| | PSATRUN #FLAG X'40' | ON IF TRUNCATED OPTION |
| 000E1F | PSATRUNI DS OXL1 | |
| | 00040 PSATRUNM EQU X'40' | |
| | PSADIVR #FLAG X'20' | ON IF DIVIDE WITH REMAINDER |
| 000E1F | PSADIVRI DS OXL1 | |
| | 00020 PSADIVRM EQU X'20' | |
| | PSATRNC #FLAG X'10' | ON IF SET TRUNCATE FLAG IN XDE'S |
| 000E1F | PSATRNCI DS OXL1 | |
| | 00010 PSATRNCM EQU X'10' | |
| | PSADIVG #FLAG X'08' | ON IF DIVIDE GIVING *LCB84097* |
| 000E1F | PSADIVGI DS OXL1 | |
| | 00008 PSADIVGM EQU X'08' | |
| | PSALLOW #FLAG X'04' | ON IF DIVIDE GIVING **R15.0** |
| 000E1F | PSALLOWI DS OXL1 | |
| | 00004 PSALLOWM EQU X'04' | |
| 000E1F | PSAASGF DS X | FLAG FOR ASSIGNMENT CMDS |
| | 00060 PSAOLEN2 EQU ((*-PSA02+3)/4)*4 | ONLINE PSA FIXED LENGTH PORTION *GWG* |
| | *** THE FOLLOWING STORAGE IS USED BY THE BATCH GENERATOR PARSER *** | PARMS FOR IDMSUTIO |
| 000E20 | PSAIOPRM DS OF | |
| 000E20 | PSAIOP1 DS A | |
| 000E24 | PSAIOP2 DS A | |
| 000E28 | PSAIOP3 DS A | |
| 000E2C | PSAIOP4 DS A | |
| 000E30 | PSAFUNC DS CL4 | FUNCTION CODE |
| 000E34 | PSASTAT DS CL4 | STATUS, RETURN CODE, OR ABEND CODE |
| 000E38 | PSAFILE DS CL4 | FILE NAME |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|-------------------------------------|
| 000E3C | PSAUTIOA DS A | ADDR OF IDMSUTIO |
| 000E40 | PSAPRTLA DS A | ADDR OF PRINT LINE ROUTINE |
| 000E44 | PSAPERRA DS A | ADDRESS OF ERROR PRINT ROUTINE |
| | PSARDIN #FLAG X'80' | ON IF READ INPUT FROM SYSIPT |
| 000E48 | PSARDINI DS 0XL1 | |
| 00080 | PSARDINM EQU X'80' | |
| | PSALIST #FLAG X'40' | ON IF WRITE OUTPUT TO SYSOUT |
| 000E48 | PSALISTI DS 0XL1 | |
| 00040 | PSALISTM EQU X'40' | |
| | PSANOPR #FLAG X'20' | ON IF LINE ALREADY PRINTED |
| 000E48 | PSANOPRI DS 0XL1 | |
| 00020 | PSANOPRM EQU X'20' | |
| | SEQCHK #FLAG X'10' | ON IF SEQUENCE CHECKING IN EFFECT |
| 000E48 | SEQCHKI DS 0XL1 | |
| 00010 | SEQCHKM EQU X'10' | |
| | PSACASE #FLAG X'08' | ON IF READ CASE SCRATCH *RZM870413* |
| 000E48 | PSACASEI DS 0XL1 | |
| 00008 | PSACASEM EQU X'08' | |
| | DS X | READ INPUT/WRITE OUTPUT FLAG |
| | *** THE FOLLOWING STORAGE IS USED BY THE ADS BATCH GENERATOR | |
| 000E4C | DS 0F | |
| 000E4C | PSAERCOD DS F | ERROR CODE FOR WTL |
| 000E50 | PSARETRN DS A | RETURN REG FOR BATCH GENERATOR |
| 000E54 | PRNTHR12 DS F | SAVE REG AREA FOR PRINT ROUTINE |
| 000E58 | PSALNLMT DS H | TOTAL LINES PER OUTPUT PAGE |
| 000E5A | PSALNCNT DS H | LINE COUNT FOR PAGE |
| 000E5C | PSAPGCNT DS H | PAGE COUNTER |
| | PSADNB #FLAG X'80' | ON IF DNB'S MUST BE BUILT |
| 000E5E | PSADNBI DS 0XL1 | |
| 00080 | PSADNBM EQU X'80' | |
| | PSAFLSH #FLAG X'40' | ON IF SIGNON BIND FAILED |
| 000E5E | PSAFLSHI DS 0XL1 | |
| 00040 | PSAFLSHM EQU X'40' | |
| | PSARGEN #FLAG X'20' | ON IF DIALOG REGENERATED |
| 000E5E | PSARGENI DS 0XL1 | |
| 00020 | PSARGENM EQU X'20' | |
| | PSADFND #FLAG X'10' | DIALOGS FOUND MEETING CRITERIA |
| 000E5E | PSADFNDI DS 0XL1 | |
| 00010 | PSADFNDM EQU X'10' | |
| | PSAPARM #FLAG X'08' | ON IF PARM FOR BGEN MSG *MET87286* |
| 000E5E | PSAPARMI DS 0XL1 | |
| 00008 | PSAPARMM EQU X'08' | |
| 000E5E | PSABTFLG DS X | BATCH GENERATOR FLAG BYTE |
| | *** THE FOLLOWING ARE PRINT LINE LAYOUTS FOR THE BATCH GENERATOR | |
| 000E60 | DS 0F | |
| 000E60 | PSAPRTLN DS OCL133 | PRINT LINE BUFFER AREA |
| 000E60 | DS C' ' | CARRIAGE CONTROL CHARACTER |
| 000E61 | PSABEGLN DS CL2 | START OF PRINT LINE |
| 000E63 | PSASTMT# DS CL5 | STATEMENT NUMBER |
| 000E68 | DS CL7 | |
| 000E6F | PSAINTXT DS CL80 | INPUT STATEMENT TEXT |
| 000EBF | DS CL28 | |
| 000EDB | ORG PSABEGLN | INPUT MESSAGE LINE LAYOUT |
| 000E61 | DS CL10 | |
| 000E6B | PSAEFLAG DS CL3 | ERROR MESSAGE INDICATOR '***' |
| 000E6E | DS CL8 | |
| 000E76 | PSAMTEXT DS CL80 | MESSAGE TEXT AREA |
| 000EC6 | DS CL32 | |
| 000EE6 | ORG PSAMTEXT | ERROR MESSAGE AREA |
| 000E76 | PSAPRCOD DS CL6 | ERROR MESSAGE CODE |
| 000E7C | DS CL3 | |
| 000E7F | PSAETEXT DS CL80 | ERROR MESSAGE TEXT AREA |
| 000ECF | DS CL24 | |
| 000EE7 | ORG PSAETEXT | PROCESS STATEMENT PRINT AREA |
| 000E7F | DS CL5 | |
| 000E84 | PSAPRTXT DS CL80 | DIALOG PROCESS PRINT AREA |

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|-----------------------|
| 000ED4 | DS | CL19 |
| 000EE8 | DS | 0F |
| 000EE8 | PSAPRHD2 | DS C' ' |
| 000EE9 | DS | CL110 |
| 000F57 | PSAPDATE | DS CL8 |
| 000F5F | DS | CL2' ' |
| 000F61 | PSAPTIME | DS CL6 |
| 000F67 | DS | CL1' ' |
| 000F68 | PSAPPAGE | DS CL5 |
| 000F6E | BFSTCOLM | DS H |
| 000F70 | BLSTCOLM | DS H |
| 000F72 | SEQSTRT | DS H |
| 000F74 | SEQLNTH | DS H |
| 000F76 | PREVSEQ | DS CL80 |
| 000FC8 | PSACNODA | DS A |
| 000FCC | PSABR13 | DS F |
| 000FD0 | DS | 10F |
| 00FF8 | PSALEN | EQU ((*-PSA+3)/4)*4 |
| 000FF8 | PSASTACK | DS OF |
| | END | START OF PARSER STACK |
| SYSTEM: | ME: WYM\$ADS | STEPNAME: ADS01 |
| Datasets | | PROCSTEP: C |
| Con DDn | | |
| P1 SYS | D 0000101.? | Volume |
| L1 SYS | R | RIG008 |
| L2 | | MVR2AB |
| SYS | O \$ADS.OBJ.H03 | |
| SYS | D 0000102.? | |
| 756K | r 261K would be required for this to be an In-Storage Assembly | |
| 9 | d 1260 Library Records Read | 0 Work File Reads |
| 0 | d 1765 Primary Print Records Written | 0 Work File Writes |
| 0 | e 0 ADATA Records Written | |
| Assembly | 1 10.30.16 Processor Time: 00.00.00.0296 | |
| Return C | | |
| OS/390 V | : Y OCTOBER 23, 2001 | |
| BATCH EM |) PGM= HEWL PROCEDURE(L) | |
| IEW2278I | R P,LET,LIST,NCAL | |
| IEW2230S | | |
| IEW2677S | NOT BE DETERMINED. | |

* M A P ***

M

DDNAME N

A P ***

PROCESSI

| | |
|-------|---|
| ALIAS | |
| ALIGN | |
| AMODE | E |
| CALL | |
| CASE | R |
| COMPA | E |

Offset Value

| | |
|-------|---|
| DCBS | |
| DYNAM | |
| EXTAT | E |
| EXITS | |
| FILL | |
| GID | E |
| HOBSE | |
| LET | |
| LINEC | |
| LIST | A |
| LISTP | |
| MAP | |
| MAXBL | 6 |
| MSGLE | |
| OVLY | |
| PRINT | |
| RES | |
| REUSA | E |
| RMODE | E |
| STORE | |
| TERM | |
| UID | E |
| UPCAS | |
| WKSPA | 0 |
| XCAL | |
| XREF | |
| ***EN | |

SAVE OPE

| | |
|-------|----------------------------|
| MEMBE | |
| LOAD | 1 RA000.WYM\$ADS.GOSET.H03 |
| PROGR | |
| VOLUM | 4 |
| MAX B | 0 |
| DISPO | D |
| TIME | 0 2001 |

SAVE MOD

| | |
|-------|---|
| AC | |
| AMODE | |
| DC | |
| EDITA | |
| EXCEE | |
| EXECU | |
| MIGRA | |
| OL | |
| OVLY | |
| PACK, | 0 |
| PAGE | |
| REFR | |
| RENT | |
| REUS | |
| RMODE | |
| SCTR | |
| SSI | |
| SYM G | |
| TEST | |

Offset ValueXPLIN
MODUL 0

ENTRY P A

NAME: E SET CLASS NAME STATUS

\$UNLABLE 000

* F R E P O R T *****

OS/390 V : Y OCTOBER 23, 2001
BATCH EM) PGM= HEWL PROCEDURE(L)
IEW2008I M URN CODE = 12.-----
MESSAGESEVERE V
2230 2ERROR M V
NONEWARNING V
NONEINFORMA V
2008 2

**** EN Y

1.102 #PWADS

```
COPY #PWADS
*****
***          PWA: ADS/ONLINE GENERATOR PROCESS WORK AREA
***          PWA IS A DSECT THAT DEFINES THE PROCESS WORK AREA FOR THE
***          ADS GENERATOR. THIS KEPT STORAGE IS MANAGED LIKE THE GWA
***          BY THE ALCSTG ROUTINE IN ADSOGNRC. THIS WORK AREA IS USED
***          TO BUILD THE GCM'S ASSOCIATED WITH A PROCESS.
***          NOTE 1: USUALLY THE GWA DSECT IS USED TO REFERENCE THE PWA
***          EXCEPT AT CLEANUP TIME.
***          NOTE 2: THIS DSECT SHOULD BE KEPT IN SYNC WITH THE GWA DSECT.
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|------------------------------|----------------------------------|
| 000000 | PWA DSECT | 14:31:01 06/15/84 |
| 000000 | PWAID DS CL4 | "PWA*" |
| 000004 | PWANXTA DS A | ADDRESS OF NEXT PWA |
| 000008 | ORG PWANXTA | |
| | PWAPRIM #FLAG X'80' | ON IF PRIMARY PWA |
| 000004 | PWAPRIMI DS 0XL1 | |
| 000080 | PWAPRIMM EQU X'80' | |
| 000004 | DS X | |
| 000005 | ORG | |
| 000008 | PWAPREVA DS A | ADDR OF PREVIOUS PWA |
| 00000C | PWASIZE DS F | SIZE OF THIS PWA |
| 000010 | PWFREEA DS A | ADDR OF 1ST FREE SPACE*RQE84167* |
| 00014 | PWALEN EQU ((*-PWA+3)/4)*4 | LENGTH OF PWA HEADER PORTION |

1.103 #RACTDS

```

COPY #RACTDS
*****
***          RACT: RCD ACTIVITY CONTROL BLOCK
***          RACT IS A DSECT THAT DESCRIBES THE ACTIVITY IN A CERTAIN ***
***          RECORD. IT IS BUILT IN ADSOGNP1, ADSOBGPS, AND ADSOBGPL ***
***          BEFORE THE CALL TO ADSOGEXC. IT IS ANCHORED OFF THE GWA ***
***          BY GWARACTA .
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|--------------------------------------|
| 000000 | RACT DSECT | 00:03:00 11/09/84 |
| 000000 | RACTNDX DS H | RECORD INDEX SAME AS GRDINDX |
| 000002 | DS H | RESERVED |
| 000004 | RACTNXTA DS A | ADDR OF NEXT RACT |
| 000008 | RERACNT DS F'00020000' | ERASE VALUE & COUNT |
| 000004 | RACTELEN EQU *-RERACNT | LENGTH OF ONE ENTRY *PHH84310* |
| 00000C | RFNDCNT DS F'00030000' | FIND VALUE & COUNT |
| 000010 | RGETCNT DS F'00050000' | GET VALUE & COUNT |
| 000014 | RCONCNT DS F'00070000' | CONNECT VALUE & COUNT |
| 000018 | RMODCNT DS F'00080000' | MODIFY VALUE & COUNT |
| 00001C | RDISCNT DS F'00110000' | DISCONNECT VALUE & COUNT |
| 000020 | RSTRCNT DS F'00120000' | STORE VALUE & COUNT |
| 000024 | ROBTCNT DS F'00430000' | OBTAIN VALUE & COUNT |
| 000028 | ROBKPCNT DS F'00630000' | OBTAIN KEEP VALUE & COUNT |
| 00002C | RFNKPCTN DS F'00230000' | FIND KEEP VALUE & COUNT |
| 000030 | RKEEPCTN DS F'00060000' | KEEP OR KEEP LONG & COUNT *PHH84313* |
| 000034 | RACCPCTN DS F'00150000' | ACCEPT VALUE & COUNT *PHH84310* |
| 000038 | RBINDCNT DS F'00140000' | RECORD BIND VALUE & COUNT *PHH84312* |
| 0003C | RACTLEN EQU *-RACT | LENGTH OF RACT |
| 0000D | RACTNENT EQU (*-RERACNT)/RACTELEN # OF ENTRIES | *PHH84310* |

1.104 #RBBDS

```

COPY #RBBDS
*****
*** RBB IS A DSECT THAT DEFINES THE RECORD BUFFER BLOCK USED BY ***
*** THE ADSO RUNTIME SYSTEM. THERE CAN BE ONE PRIMARY RBB. ***
*** SECONDARY RBB'S MAY BE CREATED WHEN THE RECORD BUFFER ***
*** NEEDS EXCEED THE CAPACITY OF THE PRIMARY RBB. THE SIZE OF ***
*** THE PRIMARY AND SECONDARY RBB'S ARE SPECIFIED BY THE USER ***
*** AT SYSGEN TIME. THE ADDRESS OF THE PRIMARY RBB IS ***
*** MAINTAINED IN THE OTB. ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|---|
| 000000 | RBB DSECT | 02/28/90 14:45:11 |
| 000000 | RBBID DS CL4 | RBB HEADER "RBB*" |
| 000004 | RBBNXTA DS A | ADDR OF NEXT RBB IN CHAIN |
| 000008 | RBBPREVA DS A | ADDR OF PREVIOUS RBB IN CHAIN |
| 00000C | RBBNABA DS A | ADDR OF NEXT AVAILABLE BUFFER |
| | RBBPRIM #FLAG X'80' | INDICATES PRIMARY RBB |
| 000010 | RBBPRIMI DS 0XL1 | |
| 000080 | RBBPRIMM EQU X'80' | |
| 000010 | RBBFLAG DS X | |
| 000011 | ORG RBBFLAG | |
| 000010 | RBBPOLSZ DS F | SIZE OF THIS BUFFER POOL |
| 000014 | RBBCOMSZ DS F | SIZE OF COMPRESSED POOL |
| | ** | EXCLUSIVE OF RBBDS |
| 000018 | RBBINDEX DS X | INDEX OF THIS RBB |
| 000019 | ORG RBBINDEX | |
| 000018 | RBBFRESZ DS F | AVAILABLE SPACE IN POOL |
| 000020 | RBBRBE DS 0D | ASSURE RBE DOULEWORD ALIGNMENT |
| 00020 | RBBLEN EQU *--RBB | LENGTH OF FIXED PORTION OF RBB |
| | * | |
| | * | * |
| | * THE RBE DSECT DEFINES THE RECORD BUFFER ELEMENT WHICH IDENTIFIES * | * |
| | * THE RECORD BUFFERS IN THE RBB. | * |
| | * | * |
| | * . FOR STANDARD RECORDS, THE RBE IMMEDIATELY PRECEDES THE BUFFER. * | * |
| | * . FOR LOGICAL RECORDS, THE LR RBE AND ONE RBE FOR EACH SUBSCHEMA * | * |
| | * RECORD WHICH PARTICIPATES IN THE LOGICAL RECORD PRECEDE THE * | * |
| | * BUFFER FOR THE ENTIRE LOGICAL RECORD. | * |
| | * . FOR RECORDS PASSED FROM A USER PROGRAM, THE RBE STANDS ALONE * | * |
| | * IN THE RBB AND CONTAINS THE ADDRESS OF THE RECORD WHICH WAS * | * |
| | * PASSED TO US FROM THE USER PROGRAM. THE RECORD BUFFER ITSELF * | * |
| | * IS NOT IN THE RBB. | * |
| | * | * |
| | * | * |
| 000000 | RBE DSECT | A |
| 000000 | RBEID DS CL4 | RBE HEADER "RBE*" 88-11-1112 |
| 000004 | RBNENAME DS CL32 | NAME OF RECORD |
| 000024 | RBERECL DS H | LENGTH OF RECORD (ROUNDED UP TO AN EVEN MULTIPLE OF 8) |
| | * | |
| 000026 | RBEVER DS H | RECORD VERSION NUMBER |
| | RBELR #FLAG X'80' | RBE FOR LOGICAL RECORD |
| 000028 | RBELRI DS 0XL1 | |
| 000080 | RBELRM EQU X'80' | |
| | RBEPSR #FLAG X'40' | RBE FOR LR ELEMENT |
| 000028 | RBEPSRI DS 0XL1 | |
| 000040 | RBEPSRM EQU X'40' | |
| | RBEPASD #FLAG X'20' | RBE FOR PASSED RECORD |
| 000028 | RBEPASDI DS 0XL1 | |

| | | | | | |
|--------|----------|---------|-------|----------------------------------|--|
| 000020 | RBEPSADM | EQU | X'20' | | |
| 000028 | RBEFLAG | DS | X | FLAG BYTE | |
| 000029 | | DS | XL3 | RESERVED | |
| 00002C | RBEVREID | DS | OF | VREID FOR RUN3.ALRECS*PHH86178* | |
| 00002C | RBELINKS | DS | H | . OLD VDB'S OTLINKS *PHH86178* | |
| 00002E | RBERDEIX | DS | H | . OLD RDE'S RDEINDX *PHH86178* | |
| 000030 | RBERECLU | DS | H | . UNROUNDED REC LEN *PHH86227* | |
| 000032 | | DS | H | . TO DBL WORD ALIGN REC ECM86220 | |
| 000034 | RBEREC | DS | OF | START OF NON-LR RECORD | |
| 000034 | 00034 | RBELEN | EQU | *-RBE | LENGTH OF FIXED PORTION OF RBE |
| 000034 | | RBEBUFA | DS | A | OFFSET TO LR RECORD/ELEMENT OR ADDRESS OF PASSED RECORD |
| | | * | | | LENGTH OF LR ELEMENT RBE |
| 000038 | RBEPSRLN | EQU | *-RBE | LOGICAL RECORD'S SUBSCHEMA | |
| 000040 | RBESUBSC | DS | CL8 | NUMBER OF ELEMENTS IN LR | |
| 000042 | RBENLRE | DS | H | RESERVED | |
| 000044 | | DS | H | LENGTH OF LOGICAL RECORD RBE | |
| 000044 | RBELRLEN | EQU | *-RBE | | |

1.105 #RDBDS

```
COPY #RDBDS
*****
***          RDB : ADS REQUEST DESCRIPTOR BLOCK
***          RDB IS A DSECT THAT DEFINES THE REQUEST DESCRIPTOR BLOCK
***          BUILT BY THE ADS BATCH GENERATOR FOR EACH DIALOG NAME
***          REQUEST IN THE INPUT STREAM.
***          "RDB*"
*****
```

Offset Value

| | | | |
|--------|----------|---------------------|-------------------------------|
| 000000 | RDB | DSECT | 16:19:09 06/15/82 |
| 000000 | RDBID | DS CL4 | "RDB*" |
| 000004 | RDBNEXTA | DS A | ADDRESS OF NEXT RDB |
| 000008 | RDBNAME | DS CL8 | DIALOG NAME |
| 000010 | RDBNAME2 | DS CL8 | DIALOG NAME FOR RANGE REQUEST |
| | RDBMSK1 | #FLAG X'80' | ON IF RDBNAME IS A MASK |
| 000018 | RDBMSK1I | DS 0XL1 | |
| 00080 | RDBMSK1M | EQU X'80' | |
| | RDBMSK2 | #FLAG X'40' | ON IF RDBNAME2 IS A MASK |
| 000018 | RDBMSK2I | DS 0XL1 | |
| 00040 | RDBMSK2M | EQU X'40' | |
| 000018 | RDBFLAG1 | DS X | FLAG BYTE |
| 0001C | RDBLEN | EQU ((*-RDB+3)/4)*4 | LENGTH OF RDB IN BYTES |
| 00007 | RDBLENF | EQU ((*-RDB+3)/4) | LENGTH OF RDB IN WORDS |

1.106 #RDEDS

| COPY #RDEDS | | | | | |
|---|--------------|-------------|--|---|----------|
| ***** RDE IS A DSECT THAT DEFINES AN ENTRY IN THE RECORD TABLE OF *** | | | | | |
| *** THE FIXED DIALOG BLOCK (FDB) OF THE ADSO RUNTIME SYSTEM. *** | | | | | |
| ***** | | | | | |
| <u>Offset</u> | <u>Value</u> | | | | |
| 000000 | RDE | DSECT | | 04/19/88 10:32:01 | 03/17/92 |
| | | PRINT NOGEN | | 14:47:01 01/04/84 | |
| 000000 | RDENAME | DS CL32 | | NAME OF RECORD | |
| 000020 | RDENRDEA | DS F | | OFFSET OF NEXT RDE | |
| 000024 | RDERECL | DS H | | LENGTH OF RECORD (EXCEPT LR'S) | |
| 000026 | ORG | RDERECL | | | |
| 000024 | RDENLRE | DS H | | # OF LOGICAL RECORD ELEMENTS | |
| 000026 | RDEVER | DS H | | RECORD VERSION NUMBER | |
| 000028 | RDEINDX | DS H | | RELATIVE VDB INDEX ENTRY | |
| 00002A | RDEMINDX | DS H | | MAP RECORD INDEX | |
| | RDEMAP | #FLAG X'80' | | RECORD USED IN ONLINE MAP | |
| | RDEPRMP | #FLAG X'40' | | RECORD USED IN PRE-MAP PROCESS | |
| | RDERESP | #FLAG X'20' | | RECORD USED IN RESPONSE | |
| | RDERVDB | #FLAG X'10' | | RECORD IS THE VDB | |
| | RDERIRA | #FLAG X'08' | | RECORD IS THE IRA | |
| | RDERLTP | #FLAG X'04' | | RECORD IS THE LITPOOL | |
| | RDEBIND | #FLAG X'02' | | RECORD MUST BE BOUND | |
| | RDERVRE | #FLAG X'01' | | RECORD IS THE VRE | |
| 00002C | RDEFLG1 | DS X | | | |
| | RDEFRSH | #FLAG X'80' | | REQUIRES NEW COPY IN BUFFER | |
| | RDELRL | #FLAG X'40' | | RDE FOR LOGICAL RECORD | |
| | RDEPSR | #FLAG X'20' | | RDE FOR LR ELEMENT | |
| | RDEPUP | #FLAG X'10' | | RECORD IS PASSED FROM USER PGM | |
| | RDEWORK | #FLAG X'08' | | RDE FOR WORK RECORD | |
| | RDEROTB | #FLAG X'04' | | RECORD IS THE OTB | |
| | RDERMSW | #FLAG X'02' | | RECORD IS THE MAP SWITCH REC | |
| | RDENEWF | #FLAG X'01' | | ON IF NEW FORMAT RDE | |
| 00002D | RDEFLG2 | DS X | | | |
| 00002E | RDECRL | DS H | | COMPRESSED INIT REC SIZE | |
| 000030 | RDEOINTV | DS OF | | START OF OLD FORMAT INIT REC | |
| 000030 | RDEINTOF | DS H | | OFFSET WITHIN RDE TO RDEINITV | |
| 000034 | RDENLRA | DS A | | OFFSET TO NEXT LOGICAL REC RDE | |
| | RDEROLE | #FLAG X'80' | | ON IF THIS IS A ROLE | |
| | * | | | OR WORK RECORD ELEMENT*MCM87023* | |
| | * | | | OF A LOGICAL RECORD *MCM87023* | |
| | RDEIMAP | #FLAG X'40' | | RECORD USED IN BATCH INPUT MAP | |
| | RDEOMAP | #FLAG X'20' | | RECORD USED IN BATCH OUTPUT MP | |
| | RDERPBF | #FLAG X'10' | | RECORD IS PBUFF (SQL) *LMA88039* | |
| | RDERDBF | #FLAG X'08' | | RECORD IS DBUFF (SQL) *LMA88039* | |
| | RDERSA | #FLAG X'04' | | RECORD IS SCA (SQL) *LMA88039* | |
| | RDEINTR | #FLAG X'02' | | INTERNAL RECORD *JMA90352* | |
| | RDESQLT | #FLAG X'01' | | CREATD FROM SQL TABLE *JMA91065* | |
| 000038 | RDEFLG3 | DS X | | FLAG BYTE | |
| | RDESR | #FLAG X'80' | | ON IF USED IN SEND/RECEIVE | |
| 000039 | RDEFLG4 | DS X | | FLAG BYTE | |
| 00003A | RDEIMNDX | DS H | | INPUT MAP RECORD INDEX | |
| 00003C | RDEOMNDX | DS H | | OUTPUT MAP RECORD INDEX | |
| 00003E | | DS H | | RESERVED | |
| 000040 | RDESCHML | DS H | | LENGTH OF SCHEMA NAME *JMA91067* | |
| | * | | | WHEN RDE CREATED FROM AN SQL TABLE *JMA91067* | |
| 000042 | RDESCHMO | DS H | | OFFSET FROM START OF RDE OF SCHEMA *JMA91067* | |
| | * | | | NAME WHEN CREATED FROM AN SQL TABLE*JMA91067* | |
| 000044 | | DS 4F | | RESERVED *JMA91067* | |
| 00054 | RDELEN | EQU *--RDE | | LENGTH OF RDE HEADER | |

PRINT GEN
00054 RDEINITV EQU * START OF COMPRESSED INIT REC

1.107 #REXDS

```
COPY #REXDS
*****
***   REX IS A CONTROL BLOCK USED BY EXECUTABLE CODE AT RUN- ***
***   TIME. REGISTER 10 WILL POINT TO REX WHEN CONTROL IS    ***
***   GIVEN TO THE EXECUTABLE CODE.                         ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|-------------------|---|
| 000000 | REX | DSECT | RUNTIME E05/26/88 16:23:58 |
| 000000 | DS | CL4 | EYECATCHER 'REX*' |
| 000004 | REXF1 | DS F | FULLWORD WORK FIELD |
| 000008 | REXD1 | DS D | Doubleword work field 1 always available for immediate use in an operation. |
| * | * | | * |
| 000010 | REXD2 | DS D | Doubleword work field 2 always available for immediate use in an operation. |
| * | * | | * |
| 000018 | REXF2 | DS F | FULLWORD WORK FIELD |
| 00001C | REXTGT | DS A | TARGET FIELD ADDRESS |
| 000020 | REXTLN | DS F | TARGET FIELD LENGTH |
| 000024 | REXSRC | DS A | SOURCE FIELD ADDRESS |
| 000028 | REXSLN | DS F | SOURCE FIELD LENGTH |
| 00002C | REXSV | DS 16F | REGISTER SAVE AREA (0 - 15) |
| 00006C | REXR10 | DS F | REGISTER 10 SAVE AREA |
| 000070 | REXVDB | DS A | VDB ADDRESS |
| 000074 | REXVRE | DS A | VRE ADDRESS |
| 000078 | REXPNC | DS A | PROCESS (RSE/PME) ADDRESS |
| 00007C | REXFDB | DS A | FDB ADDRESS |
| 000080 | REXOTB | DS A | OTB ADDRESS |
| 000084 | REXOWA | DS A | OWA ADDRESS |
| 000088 | REXCEP1 | DS A | ADDRESS OF CEXCEP1 |
| 00008C | REXCEP2 | DS A | ADDRESS OF CEXCEP2 |
| 000090 | REXERR | DS A | A(ERROR ROUTINE) |
| 000094 | REXRNC | DS F | RETURN CODE |
| 000098 | REXR13 | DS A | REGISTER 13 SAVE AREA |
| 00009C | | DS A | RESERVED |
| 0000A0 | REXMASK | DS X | PROGRAM MASK BEFORE EXECUTING CODE |
| | REXABE | #FLAG X'80' | ABEND IN PROGRESS |
| 0000A1 | REXABEI | DS 0XL1 | |
| | 00080 | REXABEM EQU X'80' | |
| 0000A1 | REXFLG1 | DS X | RUNTIME FLAG BYTE |
| 0000A2 | REXFLG2 | DS X | RUNTIME FLAG BYTE |
| | 00080 | REXTSGN EQU X'80' | TARGET IS SIGNED (IN EXECUTBL CODE) |
| 0000A3 | | DS X | RESERVED |
| 0000A4 | REXPSSA | DS F | Next instr. addr. from *MCM88147* PSW at time of pgm chk *MCM88147* |
| | * | | |
| 0000A8 | REXINTC | DS XL2 | PGM CHECK INTERRUPT CODE |
| 0000AA | REXILC | DS X | PGM CHECK INSTRUCTION LENGTH CODE |
| 0000AB | REXPMMC | DS X | PGM CHECK PROG MASK & COND CODE |
| 0000AC | REXGPN | DS A | ADDRESS OF GTOPNAD RTNE IN CEXC |
| 0000B0 | REXGPNNA | DS A | ADDRESS OF GTOPNADA RTNE IN CEXC |
| 0000B4 | RENXNTC | DS F | OFFSET FROM 1ST CMD IN PROCESS TO NEXT CMD TO BE EXECUTED RESULTING FROM A CONDITNL TEST (IF A = B ...) |
| | * | | |
| 0000B8 | REXCLUP | DS A | ADDRESS OF INSTRUCTION CLEANUP RTNE |
| 0000BC | REXCLUPA | DS A | ALTERNATE INSTRUCTION CLEANUP RTNE |
| 0000C0 | REXMVAS | DS A | MOVE ALL 'STRING' ROUTINE |
| 0000C4 | REXMVPK | DS A | MOVE PACKED DECIMAL DATA ROUTINE |
| 0000C8 | REXMVFH | DS A | MOVE Fullword to Halfword routine |
| 0000CC | REXMVBP | DS A | MOVE BINARY TO PACKED DATA ROUTINE |
| 0000D0 | REXMVPB | DS A | MOVE PACKED TO BINARY DATA ROUTINE |

| | | | | |
|---|----------|-------|---------------|--|
| 0000D4 | REXMVAP | DS | A | MOVE AFTER ADDING PACKED FIELDS |
| 0000D8 | REXMVSP | DS | A | MOVE AFTER SUBTRACTING PACKED FLDS |
| 0000DC | REXMVAB | DS | A | ADD TWO BINARY FIELDS |
| 0000E0 | REXMVSB | DS | A | SUBTRACT TWO BINARY FIELDS |
| 0000E4 | REXADBP | DS | A | ADD BINARY TO PACKED FLD |
| 0000E8 | REXSBBP | DS | A | SUBTRACT BINARY FROM PACKED FLD |
| 0000EC | REXADPB | DS | A | ADD PACKED TO BINARY FLD |
| 0000F0 | REXSBBP | DS | A | SUBTRACT PACKED FROM BINARY FLD |
| ***** This is end of R10.2 REX. | | | | |
| ***** R10.2 dialogs are dependent upon all of the above fields | | | | |
| ***** staying at the same offset. | | | | |
| 0000F4 | REXMVFH2 | DS | A | Check for ovfl on move fwd to hwd |
| 0000F8 | REXMVASB | DS | A | Move ALL 'string' routine |
| 0000FC | REVCFC | DS | A | Compare VC,FC routine |
| 000100 | REDXBTBF | DS | A | Routine to call PROCXDE to evaluate True/False DXB |
| 000104 | REXINDEX | DS | A | Calculate address of array element |
| 000108 | RENXESQ | DS | A | Get addr of instr. foll. non-COND |
| 00010C | RENXCNP | DS | A | Get addr of instr. foll. COND in a PREMAP PROCESS |
| 000110 | RENXCNR | DS | A | Get addr of instr. foll. COND in a RESPONSE PROCESS |
| 000114 | REXBRCHP | DS | A | Get addr of target instr. of a BRANCH CME in a PREMAP PROCESS |
| 000118 | REXBRCHR | DS | A | Get addr of target instr. of a BRANCH CME in a RESPONSE PROCESS |
| 00011C | REXR14 | DS | A | Register 14 save area. |
| 000120 | | DS | A | Reserved |
| 000124 | | DS | A | Reserved |
| 000128 | | DS | A | Reserved |
| 00012C | REXSBS1 | DS | F | Subscript for 1st dimension |
| 000130 | REXSBS2 | DS | F | Subscript for 2nd dimension |
| 000134 | REXSBS3 | DS | F | Subscript for 3rd dimension |
| 000138 | | DS | F | Reserved |
| 00013C | | DS | F | Reserved |
| 000140 | | DS | F | Reserved |
| 000144 | REXHWD1 | DS | H | Lib routine result area for halfword |
| 000146 | REXHWD2 | DS | H | Lib routine result area for halfword |
| 000148 | | DS | H | Reserved |
| 00014A | | DS | H | Reserved |
| 00014C | | DS | H | Reserved |
| 00014E | | DS | H | Reserved |
| 000150 | REXTF | DS | X | Lib routine result area for true/false |
| 000151 | REXHEX40 | DS | X | Always set to X'40' for pad char |
| 000152 | | DS | X | Reserved |
| 000153 | | DS | X | Reserved |
| | REXUSH1 | #FLAG | X'80' | Use REXHWD1 for rt lib rtne result |
| 000154 | REXUSH1I | DS | 0XL1 | |
| 00080 | REXUSH1M | EQU | X'80' | |
| | REXUSH2 | #FLAG | X'40' | Use REXHWD2 for rt lib rtne result |
| 000154 | REXUSH2I | DS | 0XL1 | |
| 00040 | REXUSH2M | EQU | X'40' | |
| 000154 | REXFLAG1 | DS | X | Flag byte 1 |
| 000155 | REXFLAG2 | DS | X | Reserved |
| 000156 | REXFLAG3 | DS | X | Reserved |
| 000157 | REXFLAG4 | DS | X | Reserved |
| 000158 | REMXMS | DS | CL52 | Area used to build MXS (moved XDE stack header) and a single DXB to pass to PROCXDE for run-time evaluation. |
| | * | | | |
| | * | | | |
| | * | | | |
| ***** This is end of ADS/CICS REX. | | | | |
| ***** ADS/CICS dialogs are dependent upon all of the above fields | | | | |
| ***** staying at the same offset. | | | | |
| 0018C | REXLNGTH | EQU | *-REX | LENGTH OF REX CONTROL BLOCK |
| 00063 | REXLENF | EQU | ((*-REX+3)/4) | LENGTH OF REX CONTROL BLOCK IN WORDS |

1.108 #RKEDS

```
COPY #RKEDS
*****
*** RKE IS THE DSECT THAT DEFINES THE RESPONSE KEY ELEMENT ***
*** WHICH REPRESENTS A PROGRAM FUNCTION OR OTHER KEYBOARD KEY ***
*** SYNONYMOUS WITH A VALID RESPONSE TO A FUNCTION. NOTE THAT ***
*** THERE MAY BE MANY RKE'S FOR A GIVEN KEY AS A KEY MAY HAVE ***
*** MANY MEANINGS IN RESPONSE TO DIFFERENT FUNCTIONS WITHIN AN ***
*** APPLICATION. AN RKE IS NEVER CREATED FOR THE "ENTER" KEY. ***
***
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|-----------|-------------------------------------|
| 000000 | RKE | DSECT | 18:16:21 03/01/83 |
| 000000 | RKEAPREO | DS H | SYNONYMOUS RESPONSE (APRE) INDEX |
| 000002 | RKEAID | DS X | ATTENTION IDENTIFICATION (AID) BYTE |
| 000003 | | DS X | RESERVED |
| 000004 | | DS 2F | RESERVED |
| 0000C | RKELEN | EQU *-RKE | LENGTH OF RKE |

1.109 #RLODS

```

COPY #RLODS
*****
***      RLO: ADS BASE ADDRESS BLOCK FOR RELOCATABLE STORAGE      ***
***      RLO IS THE DSECT THAT DEFINES THE BLOCK WHICH CONTAINS THE  ***
***      BASE ADDRESSES FOR ADS RELOCATABLE STORAGE BLOCKS. WHEN      ***
***      "RESOURCES ARE RELOCATABLE" IS SPECIFIED IN THE ADSO      ***
***      STATEMENT, THIS IS THE ONLY RUN TIME STORAGE THAT IS KEPT      ***
***      ACCROSS THE PSEUDO-CONVERSE. THIS IS WHERE ADS TELLS DC      ***
***      TO PUT THE ADDRESSES OF THE ADDRESSES OF THE RELOCATABLE      ***
***      STORAGE.                                                 ***
*****                                                       ***
Offset  Value
000000    RLO      DSECT          11:26:06 07/01/86
000000    RLOOTBA DS   A           ADDRESS OF THE OTB ADDRESS
000004    RLOOTBXA DS  A           ADDRESS OF THE OTB EXTENSION ADDRESS
000008    RLOHLMA DS  A           ADDRESS OF THE HELP STORAGE ADDRESS
00000C    RLORCBWA DS A           ADDRESS OF RECB WORKAREA ADDR *PHH86182*
000010            DS  A           RESERVED
000014    RLODLGTB DS  0F          START OF DIALOG SPECIFIC ADDRESSES
00014     RLOLEN EQU  *--RLO      LENGTH OF FIXED RLO
*
*****
***      RLOD DEFINES A DIALOG SPECIFIC REPEATABLE SECTION OF      ***
***      THE RLO.                                                 ***
*****                                                       ***
000000    RLOD      DSECT
000000    RLOVDBA DS   A           ADDRESS OF VDB ADDRESS
00004     RLODLEN EQU  *--RLOD      LENGTH OF ONE DIALOG SECTION

```

1.110 #RPEDS

| COPY #RPEDS | | | | |
|--|---------------------|--|--|--|
| ***** RPE IS A DSECT WHICH DEFINES A REPLACEMENT PARAMETER ELEMENT *** *** WHICH IS FOUND IN A DISPLAY COMMAND CME. RPE'S ARE PLACED *** *** IN THE DISPLAY CME WHENEVER THE USER REQUIRES REPLACEMENT *** *** PARAMETERS TO BE INSERTED IN HIS DISPLAYED MESSAGE. THERE IS *** *** ONE RPE FOR EACH REPLACEMENT PARAMETER. THEY ARE PLACED IN *** *** THE SAME ORDER AS REQUIRED TO BE PASSED ON VIA THE #WTL MACRO.*** ***** | | | | |
| <u>Offset</u> | <u>Value</u> | | | |
| 000000 | RPE DSECT | | 12:14:05 05/28/81 | |
| | RPENONS #FLAG X'80' | | | RPE IS FOR A NON-SUBSCRIPTED FIELD |
| 000000 | RPENONSI DS 0XL1 | | | |
| 00080 | RPENONSM EQU X'80' | | | |
| | RPESUBV #FLAG X'40' | | | RPE IS FOR A SUBSCRIPTED FIELD |
| 000000 | RPESUBVI DS 0XL1 | | | |
| 00040 | RPESUBVM EQU X'40' | | | |
| | * | X'00' | | THE ABSENCE OF AN ON BIT IN THE FLAG INDICATES A LITERAL IN THE LITPOOL |
| 000000 | RPEFLAG DS X | | | RPE FLAG BYTE |
| 000001 | RPEDLEN DS X | | | LENGTH OF REPLACEMENT TEXT (EXCEPT FOR LITERALS IN THE LITPOOL) |
| 000002 | RPEOFFS DS H | * FOR LITERALS: LITPOOL OFFSET TO REPLACEMENT TEXT | | |
| | * | * FOR NON-SUBSCRITED FIELDS: VRE TABLE OFFSET | | |
| | * | * FOR SUBSCRIPTED FIELDS: CME OFFSET NEXT RPE | | |
| 00004 | RPEF1LN EQU *-RPE | | LENGTH OF FORMAT 1 RPE (LITERAL) | |
| 00004 | RPEXDES DS 0F | | START OF XDE STACK FOR SUBSCRIPTED FLD | |
| 00004 | RPEDTAOF DS H | | DATA OFFSET TO REPLACEMENT TEXT | |
| 00006 | DS H | | UNUSED | |
| 00008 | DS 0F | | ASSURE FULLWORD ALIGNMENT OF NEXT RPE | |
| 00008 | RPEF2LN EQU *-RPE | | LENGTH OF FORMAT 2 RPE (NO SUBSCRIPT) | |

1.111 #RSEDS

```
COPY #RSEDS
*****
*** RSE IS A DSECT THAT DEFINES THE HEADER PORTION OF THE      ***
*** COMMAND TABLE WHICH COMPRISES ONE RESPONSE PROCESS FOR AN      ***
*** ADSO DIALOG.  THE RESPONSE ELEMENTS FOR A DIALOG ARE CHAINED ***
*** TOGETHER IN THE FIXED DIALOG BLOCK (FDB).                      ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--|--|
| 000000 | RSE DSECT | 15:52:13 05/23/85 |
| 000000 | RSENNAME DS CL32 | NAME OF RESPONSE |
| 000020 | RSENXTA DS F | OFFSET OF NEXT RSE IN FDB |
| 000024 | RSELASTB DS F | OFFSET OF LAST BYTE IN RSE |
| 000028 | RSERATA DS F | OFFSET TO READY AREA TABLE |
| 00002C | RSEFCMEA DS F | OFFSET TO FIRST CME |
| 000030 | RSEPVER DS H | PROCESS VERSION |
| 000032 | RSENCMES DS H | NUMBER OF CME'S IN RESPONSE |
| | * RSEPFKEY FIELD CONTAINS THE INDICATORS FOR BATCH CONTROL | |
| | * EVENTS AS WELL AS THE PROGRAM FUNCTION KEYS | |
| 000034 | RSEPFKEY DS X | PFKEY FOR RESPONSE |
| 000035 | RSEFUNLN DS X | LENGTH OF RESPONSE FIELD |
| 000036 | RSEOFUNC DS X | START OLD FORMAT FUNCTION CODE |
| | PRINT NOGEN | |
| | RSERDY #FLAG X'80' | ON IF AREAS MUST BE READIED |
| | RSEALOW #FLAG X'40' | ON IF AUTO-EDIT ERRORS ALLOWED |
| | RSERDUP #FLAG X'20' | ON IF RP DUPLICATES EXISTING RP |
| | RSERALL #FLAG X'10' | ON IF RAT HAS READY ALL BYTE*JEB84151* |
| | RSEDEF #FLAG X'08' | ON FOR THE DEFAULT RESPONSE PROCESS |
| 000037 | RSEFLAG1 DS X | |
| | PRINT GEN | |
| 000038 | RSEFUNOF DS H | OFFSET WITHIN RSE TO FUNC CODE |
| 00003A | RSENMDTE DS H | NUMBER OF MODULE TABLE ENTRIES |
| 00003C | RSELNTA DS F | OFFSET TO CME LINE NUMBER TABLE |
| 000040 | RSEDATLU DS CL8 | DATE MODULE-067 LAST UPDATED |
| 000048 | RSEDATCR DS CL8 | DATE MODULE-067 CREATED |
| 000050 | RSEMDTA DS F | OFFSET TO INCLUDE MODULE TABLE |
| 000054 | RSEOFBTBL DS F | OFFSET OF CME & EXEC CD OFFSET TBL |
| 000058 | RSEEXTA DS F | RESERVED -ADDR OF RSE EXTENSION |
| 00005C | RSELEN EQU *-RSE | LENGTH OF RSE FIXED PORTION |
| 00005C | RSEFUNC DS 0X | START OF RESPONSE FUNCTION CODE |

1.112 #RSPDS

```

COPY #RSPDS
*****
***      RSP:  ADS APPLICATION GENERATOR RESPONSE RECORD
***      RSP IS A DSECT THAT DESCRIBES THE FIELDS IN THE
***      ADSO-RESPONSE RECORD USED BY THE ADS APPLICATION GENERATOR.
*****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------|---------------------------------|
| 000000 | RSP DSECT | 11/27/89 16:15:45 05/24/91 |
| 000000 | RSPNEST DS F | NEST CODE |
| 000004 | RSPNAME DS CL8 | RESPONSE NAME |
| 00000C | RSPAID DS X | PFKEY |
| 00000D | RSPSECCL DS X | SECURITY CLASS CODE |
| 00000E | RSPDESC DS CL28 | RESPONSE DESCRIPTION |
| 00002A | RSPRTYPE DS X | RESPONSE TYPE |
| 000D3 | RSPLOCAL EQU C'L' | LOCAL |
| 000C7 | RSPGLOBL EQU C'G' | GLOBAL |
| 00002B | RSPCTRL DS X | INTERDIALOG CONTROL COMMAND |
| 000E3 | RSPTRAN EQU C'T' | TRANSFER |
| 000C9 | RSPINV EQU C'I' | INVOKE |
| 000D3 | RSPLINK EQU C'L' | LINK |
| 000D9 | RSPRETN EQU C'R' | RETURN |
| 00002C | RSPINVFC DS CL8 | INVC-FUNC NAME |
| | RSPXIMM #FLAG X'80' | ON - EXECUTE IMMEDIATE |
| 000034 | RSPXIMMI DS OXL1 | |
| | 00080 RSPXIMMM EQU X'80' | |
| | * | OFF - EXECUTE DEFERED (DEFAULT) |
| | RSPDBNS #FLAG X'40' | DB OPTION - NOSAVE |
| 000034 | RSPDBNSI DS OXL1 | |
| | 00040 RSPDBNSM EQU X'40' | |
| | RSPDBNF #FLAG X'20' | DB OPTION - NOFINISH |
| 000034 | RSPDBNFI DS OXL1 | |
| | 00020 RSPDBNFM EQU X'20' | |
| | RSPRCLR #FLAG X'08' | RETURN - CLEAR |
| 000034 | RSPRCLRI DS OXL1 | |
| | 00008 RSPRCLRM EQU X'08' | |
| | RSPRCNT #FLAG X'04' | RETURN - CONTINUE |
| 000034 | RSPRCNTI DS OXL1 | |
| | 00004 RSPRCNTM EQU X'04' | |
| 000034 | RSPFLAG1 DS XL1 | |
| 000035 | DS XL3 | RESERVED |
| 00038 | RSPLEN EQU *-RSP | LENGTH OF DSECT |
| 0000E | REPLENF EQU (*-RSP+3)/4 | LENGTH IN FULLWORDS |

1.113 #RSRTABL

```
COPY #RSRTABL
*****
*          *
* DSECT FOR RESPONSE SORT TABLE HEADER (RSRTABLE)      *
*          *
*****  


| <u>Offset</u> | <u>Value</u>                                     |                                     |
|---------------|--------------------------------------------------|-------------------------------------|
| 000000        | RSRTABLE DSECT                                   | 12:07:23 04/04/85                   |
| 000000        | RSRTNENT DS H                                    | NUMBER OF USED ENTRIES              |
| 000002        | RSRTTENT DS H                                    | NUMBER OF AVAILABLE ENTRIES         |
| 000004        | DS F                                             | RESERVED                            |
| 000008        | RSRTLEN EQU *--RSRTABLE                          | LENGTH OF RESPONSE TABLE HEADER     |
|               |                                                  | *****                               |
|               | *                                                | *                                   |
|               | * DSECT FOR RESPONSE SORT TABLE ENTRY (RSRTNTRY) | *                                   |
|               | *                                                | *                                   |
|               | *****                                            |                                     |
| 000000        | RSRTNTRY DSECT                                   |                                     |
| 000000        | RSRTNAME DS CL8                                  | RESPONSE NAME                       |
| 000008        | RSRTCKEY DS CL8                                  | RESPONSE CONTROL KEY                |
| 000010        | RSRTFNAM DS CL8                                  | RESPONSE FUNCTION NAME              |
| 000018        | RSRTINDX DS H                                    | RESPONSE FUNCTION'S INDEX           |
| 00001A        | RSRTRNDX DS H                                    | RESPONSE INDEX                      |
| 0001C         | RSRTELEN EQU *--RSRTNTRY                         | LENGTH OF RESPONSE SORT TABLE ENTRY |


```

1.114 #RSTDs

```

COPY #RSTDs
*****
*          *
* DSECT FOR RESPONSE TABLE HEADER (RSTBHDR)      *
*          *
*****




| <u>Offset</u> | <u>Value</u> |                       |                                    |                |
|---------------|--------------|-----------------------|------------------------------------|----------------|
| 000000        | RSTBHDR      | DSECT                 | 12/06/89 16:10:10                  | 03/11/93       |
| 000000        | RSTID        | DS CL4                | "RST*"                             |                |
| 000004        | RSTAPPL      | DS CL8                | APPLICATION NAME                   |                |
| 00000C        | RSTMAGE      | DS H                  | MAXIMUM NUMBER OF ENTRIES          |                |
| 00000E        | RSTTTLU      | DS H                  | TOTAL NUMBER OF ENTRIES USED       |                |
| 000010        | RSTCURRA     | DS A                  | ADDRESS OF CURRENT ENTRY           |                |
| 000014        | RSTSEQC      | DS F                  | SEQUENCE COUNTER                   |                |
|               | RSTSEQU      | #FLAG X'80'           | SEQUENCE NUMBERS HAVE BEEN UPDATED |                |
| 000018        | RSTSEQUI     | DS OXL1               |                                    |                |
|               | 00080        | RSTSEQUM EQU X'80'    |                                    |                |
|               |              | RSTSTBL #FLAG X'40'   | SEQUENCE TABLE EXISTS              |                |
| 000018        | RSTSTBLI     | DS OXL1               |                                    |                |
|               | 00040        | RSTSTBLM EQU X'40'    |                                    |                |
|               |              | RSTRSRT #FLAG X'20'   | RESORT SELECTED ENTRIES            | **ADS370/104** |
| 000018        | RSTRSRTI     | DS OXL1               |                                    |                |
|               | 00020        | RSTRSRM EQU X'20'     |                                    |                |
|               |              | *                     | TO TOP OF TABLE REQUIRED           | **ADS370/104** |
| 000018        | RSTFLAG1     | DS X                  |                                    |                |
| 000019        |              | DS X                  |                                    |                |
| 00001C        | RSTFNTRY     | DS OF                 | START OF RSTB ENTRIES              |                |
|               | 0001C        | RSTHLEN EQU *-RSTBHDR | LENGTH OF RESPONSE TABLE HEADER    |                |
| <u>Offset</u> | <u>Value</u> |                       |                                    |                |
| 000000        | RSTBNTRY     | DSECT                 | 12:58:48 04/04/85                  |                |
| 000000        | RSTRESP      | DS CL8                | RESPONSE NAME                      |                |
| 000008        | RSTAID       | DS X                  | AID BYTE                           |                |
| 000009        | RSTFUNC      | DS CL8                | FUNCTION NAME                      |                |
| 000011        | RSTRECNO     | DS XL2                | RELATIVE RECORD NUMBER (BINARY)    |                |
| 000013        |              | DS X                  |                                    |                |
| 000014        | RSTSEQNO     | DS F                  | SEQUENCE NUMBER                    |                |
| 00018         | RSTNLEN      | EQU *-RSTBNTRY        | LENGTH OF RESPONSE TABLE ENTRY     |                |


```

1.115 #RVBDS

```
COPY #RVBDS
*****
***          RVB : ADS REQUEST VERSION BLOCK
***          RVB IS A DSECT THAT DEFINES THE REQUEST VERSION BLOCK BUILT
***          BY THE ADS BATCH GENERATOR FOR EACH VERSION QUALIFICATION
***          ON AN INPUT COMMAND.
***          RVB*"
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------------|-----------------------------------|
| 000000 | RVB DSECT | 16:19:25 06/15/82 |
| 000000 | RVBID DS CL4 | "RVB*" |
| 000004 | RVBNEXTA DS A | ADDRESS OF NEXT RVB |
| 000008 | RVBVER1 DS H | FIRST VERSION NUMBER |
| 00000A | RVBVER2 DS H | SECOND VERSION (IF RANGE REQUEST) |
| 00000C | RVBLEN EQU ((*-RVB+3)/4)*4 | LENGTH OF RVB IN BYTES |
| 000003 | RVBLENF EQU ((*-RVB+3)/4) | LENGTH OF RVB IN WORDS |

1.116 #SACTDS

```

COPY #SACTDS
*****
***          SACT: SET ACTIVITY CONTROL BLOCK
***          ***
***          *** SACT IS A DSECT THAT DESCRIBES THE ACTIVITY IN A CERTAIN ***
***          SUBSCHEMA SET. IT IS BUILT IN ADSOGNP1, ADSOBGPS,
***          AND ADSOBGPL BEFORE THE CALL TO ADSOGEXC. IT IS ANCHORED ***
***          OFF OF THE GWA (BY GWASACTA).
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|-----------------------------------|--------------------------------------|
| 000000 | SACT DSECT | 18:14:13 12/06/84 |
| 000000 | SACTNAME DS CL32 | SET NAME |
| 000020 | SACTNXTA DS A | ADDRESS OF NEXT SACT |
| 000024 | SFNDCNT DS F'00030000' | FIND VALUE & COUNT |
| 00004 | SACTELEN EQU *-SFNDCNT | LENGTH OF ONE ENTRY *PHH84310* |
| 000028 | SCONCNT DS F'00070000' | CONNECT & COUNT |
| 00002C | SDISCNT DS F'00110000' | DISCONNECT & COUNT |
| 000030 | SOBTCNT DS F'00430000' | OBTAIN & COUNT |
| 000034 | SOBKPCNT DS F'00630000' | OBTAIN KEEP VALUE & COUNT |
| 000038 | SFNKPCNT DS F'00230000' | FIND KEEP VALUE & COUNT |
| 00003C | SKEEPCNT DS F'00060000' | KEEP OR KEEP LONG & COUNT *PHH84313* |
| 000040 | SACCPCNT DS F'00150000' | ACCEPT VALUE & COUNT *PHH84310* |
| 000044 | SIFCNT DS F'00160000' | IF SET VALUE & COUNT *PHH84310* |
| 000048 | SRETCNT DS F'00170000' | RETURN DBKEY & COUNT 0* |
| 0004C | SACTLEN EQU *-SACT | LENGTH OF SACT |
| 0000A | SACTNENT EQU (*-SFNDCNT)/SACTELEN | # OF ENTRIES *PHH84310* |

1.117 #SDEDS

```
COPY #SDEDS
*****
*** SDE IS A DSECT THAT DESCRIBES THE SDSE-044 RECORD ON THE ***
*** DATA DICTIONARY.                                           ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|---------|
| 000000 | SDE | DSECT |
| 000000 | SDESEQ | DS F |
| 000004 | SDEFVALN | DS H |
| 000006 | SDESVALN | DS H |
| 000008 | SDEFVAL | DS CL34 |
| 00002A | SDESVAL | DS CL34 |
| 00004C | | DS CL28 |
| 000068 | SDECID | DS F |

| | | | |
|--------|----------|---------|------------------------|
| 000000 | SDE | DSECT | 16:29:35 02/28/81 |
| 000000 | SDESEQ | DS F | LINE SEQUENCE NUMBER |
| 000004 | SDEFVALN | DS H | LENGTH OF FIRST VALUE |
| 000006 | SDESVALN | DS H | LENGTH OF SECOND VALUE |
| 000008 | SDEFVAL | DS CL34 | FIRST VALUE |
| 00002A | SDESVAL | DS CL34 | SECOND VALUE |
| 00004C | | DS CL28 | FILLER |
| 000068 | SDECID | DS F | COMMENT TYPE |

1.118 #SELDS

```

COPY #SELDS
*****
***          SELCT: SELECT TABLE STORAGE
***          SELCT IS A DSECT THAT DESCRIBES SELECT TABLE STORAGE
***          USED IN THE ADSO RUN TIME SYSTEM UNDER THE APPLICATION
***          CONTROL FACILITY. IT IS USED FOR BUILDING HELP AND
***          MENU MAPS.
***          ENTIRE SELECT TABLE
*****

```

Offset Value

| | | | |
|--------|-----------|----------------|----------------------------------|
| 000000 | SELCT | DSECT | 13:05:27 01/05/83 |
| 000000 | SELSTBL | DS 50CL38 | ENTIRE SELECT TABLE |
| 00076C | | ORG SELSTBL | |
| 000000 | SELSNTRY | DS 0CL38 | |
| 000000 | SELSLECT | DS CL1 | SELECT FLAG |
| 000001 | SELSRESP | DS CL8 | RESPONSE FIELD NAME |
| 000009 | SELSKEY | DS CL1 | KEY ASSOCIATED WITH RESPONSE |
| 00000A | SELDSDSCR | DS CL28 | FUNCTION DESCRIPTION |
| 00026 | SELSLEN | EQU *-SELSNTRY | LENGTH OF A SELECT ELEMENT ENTRY |
| 000026 | | ORG | |
| 0076C | SELSTBLN | EQU *-SELSTBL | LENGTH OF ENTIRE TABLE |
| 00032 | SELSLNUM | EQU 50 | NUMBER OF ENTRIES |

1.119 #SFEDS

```
COPY #SFEDS
*****
*** SFE IS A DSECT THAT DEFINES THE SUBSCRIPTED FIELD ELEMENT. ***
*** SFE'S ARE IN MRE INDEX LISTS WHEN SUBSCRIPTED FIELDS ARE ***
*** REFERENCED WHOSE SUBSCRIPTS CANNOT BE DETERMINED UNTIL RUN ***
*** TIME. AN SFE CONSISTS OF A HEADER PORTION FOLLOWED BY ONE ***
*** SUBSCRIPT DEFINITION ELEMENT (SDE) FOR EACH LEVEL OF ***
*** SUBSCRIPTING. ***
*** AN SFE USED IN AN MRE INDEX LIST ALWAYS POINTS TO A ***
*** SECONDARY MRE INDEX LIST WHICH CONTAINS ONE MRE INDEX FOR ***
*** EACH OCCURRENCE OF THE SUBSCRIPTED FIELD. USING THE SDE(S) ***
*** THE SUBSCRIPT VALUE(S) IS(ARE) DETERMINED. THE SUBSCRIPT ***
*** VALUE(S) IS(ARE) THEN USED TO DERIVE AN INDEX INTO THE ***
*** SECONDARY MRE INDEX LIST TO PICK THE CORRECT MRE INDEX. ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------|---|
| 000000 | SFE DSECT | 15:34:14 04/22/81 |
| | SFEID #FLAG X'80' | ALWAYS ON, IDENTIFIES AN SFE |
| 000000 | SFEIDI DS 0XL1 | |
| | 00080 SFEIDM EQU X'80' | |
| 000000 | SFEFLAG DS X | SFE FLAG BYTE |
| 000001 | SFENDIMS DS X | # OF DIMENSIONS (# OF SDE'S FOLLOWING) |
| 000002 | SFEOFFS DS H | OFFSET TO SECONDARY MRE INDEX LIST |
| | 00004 SFELEN EQU *-SFE | LENGTH OF HEADER PORTION OF SFE |
| | | ***** |
| | | *** SDE IS A DSECT THAT DESCRIBES A SUBSCRIPT DEFINITION ELEMENT *** |
| | | *** FOUND IN AN SFE. AN SFE CONTAINS ONE SDE FOR EACH DIMENSION. *** |
| | | *** SDE'S HAVE TWO FORMATS. FORMAT 1 IS USED TO POINT TO AN XDE *** |
| | | *** STACK WHICH WILL BE USED TO EVALUATE A SUBSCRIPT WHICH IS *** |
| | | *** WRITTEN AS AN EXPRESSION. FORMAT 2 CONTAINS XDE-LIKE ELEMENTS *** |
| | | *** USED TO DESCRIBE THE LOCATION AND FORMAT OF THE SUBSCRIPT. *** |
| | | ***** |
| 00006C | SDE DSECT | |
| | SDEID #FLAG X'80' | ON = FORMAT 1 SDE; OFF = FORMAT 2 SDE |
| 00006C | SDEIDI DS 0XL1 | |
| | 00080 SDEIDM EQU X'80' | |
| 00006C | SDEMAXV DS H | MAXIMUM SUBSCRIPT VALUE |
| | * | |
| | ** FORMAT 1 ** | |
| | * | |
| 00006E | SDESTKSZ DS H | XDE STK SIZE INCLUDING HEADER |
| 000070 | SDEXDESA DS A | XDE STK ADDR (FOR GENERATOR USE) |
| 000074 | ORG SDEXDESA | |
| 000070 | SDEXDEOF DS H | LITPOOL OFFSET TO XDE STACK FOR SUBSCRIPT |
| 000072 | DS H | |
| | 00074 SDEF1LEN EQU *-SDE | LENGTH OF FORMAT 1 SDE |
| | * | |
| | ** FORMAT 2 ** | |
| | * | |
| 000074 | ORG SDESTKSZ | |
| 00006E | SDEDATYP DS X | SUBSCRIPT'S DATA TYPE (AS IN XDE) |
| 00006F | SDEDATLN DS X | SUBSCRIPT'S LENGTH |
| 000070 | SDEDTABO DS H | VRE TABLE OFFSET TO SUBSCRIPT'S RECORD |
| 000072 | SDEDDSPL DS H | SUBSCRIPT'S OFFSET WITHIN THE RECORD |
| | 00074 SDEF2LEN EQU *-SDE | LENGTH OF FORMAT 2 SDE |
| | 00074 SDELEN EQU *-SDE | LENGTH OF SDE (FORMAT 1&2 ARE SAME SIZE) |

1.120 #SFTDS

```
COPY #SFTDS
*****
***   ADSA SYSTEM FUNCTION TABLE
*** -----
***   THIS DSECT DESCRIBES THE ADSA SYSTEM 12:23:37 04/04/85
***   TABLE CONTAINS ONE ENTRY FOR EACH SYSTEM DEFINED FUNCTION. ***
*****
```

| <u>Offset</u> | <u>Value</u> |
|---------------|----------------------|
| 000000 | SFT DSECT |
| 000000 | SFTNTRY5 DS H |
| 000002 | SFTNLEN DS H |
| 000004 | SFTSTART EQU * |
| 000000 | SFTE DSECT |
| 000000 | SFTEFCOD DS H |
| 000002 | SFTEFNAM DS CL8 |
| 0000A | SFTELEN EQU *-SFTE |

| | | |
|--------|----------------------|--------------------------------|
| 000000 | SFT DSECT | |
| 000000 | SFTNTRY5 DS H | ENTRIES IN TABLE |
| 000002 | SFTNLEN DS H | LENGTH OF ONE ENTRY |
| 000004 | SFTSTART EQU * | START OF SYSTEM FUNCTION TABLE |
| 000000 | SFTE DSECT | |
| 000000 | SFTEFCOD DS H | SYSTEM FUNCTION CODE |
| 000002 | SFTEFNAM DS CL8 | FUNCTION NAME |
| 0000A | SFTELEN EQU *-SFTE | LENGTH OF SYS FUNC TBL ENTRY |

1.121 #SREDS

```
COPY #SREDS
*****
***          SRE : ADSO SCREEN RECORD ELEMENT
***          SRE IS A DSECT THAT DEFINES A SCREEN RECORD ELEMENT USED
***          BY THE ADS ONLINE GENERATOR. THERE IS AN SRE FOR THE
***          DIALOG SCREEN, PREMAP SCREEN, DIALOG OPTIONS SCREEN,
***          EACH RESPONSE SCREEN, AND EACH NEW COPY SCREEN.
***          THE SRE FORMAT VARIES DEPENDING ON THE TYPE OF SCREEN
***          IT REPRESENTS.
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------------------|---------------------------------|
| 000000 | SRE DSECT | 09:57:18 07/31/86 |
| 000000 | SREID DS CL4 | "SRE*" |
| 000004 | SRENEXTA DS A | ADDRESS OF NEXT SRE |
| 000008 | SREPREVA DS A | ADDRESS OF PREVIOUS SRE |
| 00000C | SRETYPE DS H | TYPE OF SCREEN REPRESENTED |
| 00000E | SREDDNAM DS CL8 | DIALOG NAME |
| 000016 | SREDMPNM DS CL8 | MAP NAME |
| 00001E | SREDSSNM DS CL8 | SUBSCHEMA NAME |
| 000026 | SREDMDDU DS CL7 | DISPLAY/UPDATE MODE |
| 00002D | SREDMDFS DS CL4 | FAST/STEP MODE |
| 00031 | SRELEN EQU *~-SRE | LENGTH OF SRE |
| 000034 | SREDG DS 0F | DIALOG SCREEN - TYPE 01 |
| 000034 | SREDDICT DS CL8 | DICTIONARY NAME |
| 00003C | SREDNODE DS CL8 | NODE NAME |
| 000044 | SREDSCNM DS CL8 | SCHEMA NAME |
| 00004C | SREDAREC DS CL32 | AUTOSTATUS RECORD NAME |
| 00006C | SREDDVER DS H | DIALOG VERSION |
| 00006E | SREDSCVR DS H | SCHEMA VERSION |
| 000070 | SREDMPVR DS H | MAP VERSION |
| 000072 | SREDAVER DS H | AUTOSTATUS RECORD VERSION |
| 000074 | SREDACTN DS CL3 | ACTION CODE |
| 000077 | SREDMNLN DS CL3 | MAINLINE INDICATOR |
| 00007A | SREDAUTO DS CL3 | AUTOSTATUS INDICATOR |
| 00007D | SREDMESS DS 4CL79 | MESSAGE AREAS |
| 0001B9 | SREDSLCT DS 22CL1 | FUNCTION SELECT BYTES |
| 0001CF | SREDSRSP DS CL32 | SELECT RESPONSE NAME |
| 0001EF | SREDSREC DS CL32 | SELECT RECORD NAME |
| 00020F | SREDUNXF DS CL1 | UMBRELLA NEXT TASK FLAG (0/N) |
| 000210 | SREDRVER DS H | SELECT RECORD VERSION |
| 000212 | SREDPFKS DS CL2 | PF KEY SIMULATOR FIELD |
| 000214 | DS DS CL12 | RESERVED FOR FUTURE USE |
| 000220 | SREDUNXT DS CL8 | UMBRELLA NEXT TASK CODE |
| 00228 | SREDPRT1 EQU *~-SRE | LEN OF MOVE FROM SRO *JMA85064* |
| 000228 | SREBIMN DS CL8 | BATCH INPUT MAP NAME *JMA85063* |
| 000230 | SREBOMN DS CL8 | BATCH OUTPUT MAP NAME*JMA85063* |
| 000238 | SREBIMV DS H | BATCH INPUT MAP VERS *JMA85063* |
| 00023A | SREBOMV DS H | BATCH OUTPUT MAP VERS*JMA85063* |
| 00023C | DS DS X | RESERVED FOR FUTURE *JMA85063* |
| 00240 | SREDLEN EQU ((~-SRE+3)/4)*4 | LENGTH OF DIALOG SRE IN BYTES |
| 00090 | SREDLENF EQU ((~-SRE+3)/4) | LENGTH OF DIALOG SRE IN WORDS |
| 00023D | SREPR ORG SREDG | PREM - 02, RESPONSE - 03 |
| 000034 | SREPPNAM DS CL32 | PROCESS NAME |
| 000054 | SREPPVER DS H | PROCESS VERSION |
| 000056 | SREPPNLN DS H | TOTAL COMMANDS |
| 000058 | SREPPNER DS H | TOTAL NUMBER OF ERRORS |
| 00005A | SREPACTN DS CL3 | ACTION (DELETE) |
| 00005D | SREPLMS DS CL28 | TOTAL LINE MESSAGE |

| | | | |
|--------|--------------|-----------------|---------------------------------|
| 000079 | SREPPEMS DS | CL22 | ERROR MESSAGE |
| 00008F | SREPMESS DS | 4CL79 | SCREEN MESSAGES |
| 0001CB | SREPSSLCT DS | 22CL1 | SELECT BYTES |
| 0001E1 | SREPSRSP DS | CL32 | RESPONSE NAME |
| 000201 | SREPSREC DS | CL32 | RECORD NAME |
| 000221 | SREPUNXF DS | CL1 | UMBRELLA NEXT TASK FLAG (0/N) |
| 000222 | SREPVER DS | H | SELECT RECORD VERSION |
| 000224 | SREPPFKS DS | CL2 | PF KEY SIMULATOR FIELD |
| 000226 | DS | CL12 | RESERVED FOR FUTURE USE * |
| 000232 | SREPUNXT DS | CL8 | UMBRELLA NEXT TASK CODE |
| 0023A | SREPRT1 EQU | *-SRE | LEN OF MOVE FROM SRO *JMA85064* |
| | DS | CL13 | RESERVED FOR FUTURE U*JMA85064* |
| 000247 | DS | CL8 | RESERVED FOR FUTURE USE |
| 00250 | SREPLEN EQU | ((*-SRE+3)/4)*4 | LENGTH OF PREMAP SRE IN BYTES |
| 00094 | SREPLENF EQU | ((*-SRE+3)/4) | LENGTH OF PREMAP SRE IN WORDS |
| 000250 | DS | OF | |
| 000250 | SREGPRA DS | A | ADDRESS OF CORRESPONDING GPR |
| 000254 | SREPCKY DS | CL5 | RESPONSE CONTROL KEY |
| 000259 | SREPFRNC DS | CL32 | RESPONSE FUNCTION CODE |
| 000279 | SREPREDT DS | CL3 | ALLOW AUTO-EDIT ERRORS (YES,NO) |
| 0002C | SREPRT2 EQU | *-SREGPRA | LEN OF MOVE FROM SRO TO SRE |
| 00027C | SREPRBCE DS | CL6 | BATCH CNTRL ELEMENT *JMA85064* |
| 000282 | SREPDEF DS | C | DEFAULT RESPONSE IND.*JMA85064* |
| 00284 | SERLEN EQU | ((*-SRE+3)/4)*4 | LENGTH OF RESPONSE IN BYTES |
| 000A1 | SERLENF EQU | ((*-SRE+3)/4) | LENGTH OF RESPONSE IN WORDS |
| 000283 | SRENCP ORG | SREDG | NEW COPY SCREEN - TYPE 04 |
| 000034 | SRENNCPY DS | OCL320 | NEW COPY RECORD AREA *JMA85064* |
| 000034 | SRENNCLN DS | H | LINE NUMBER |
| 000036 | SRENNCNM DS | CL32 | NEW COPY RECORD NAME |
| 000056 | SRENNCVR DS | H | RECORD VERSION |
| 000058 | SRENNCNC DS | CL1 | NEW COPY INDICATOR |
| 000059 | SRENNCWK DS | CL1 | WORK RECORD INDICATOR |
| 00026 | SRENPRT1 EQU | *-SRENNCLN | LEN OF MOVE FROM SRO *JMA85064* |
| 00005A | DS | CL1 | RESERVED FOR FUTURE * |
| 00005B | DS | CL1 | RESERVED FOR FUTURE *JMA85064* |
| 00005C | DS | 7CL40 | *JMA85064* |
| 000174 | SRENMESS DS | 4CL79 | SCREEN MESSAGES |
| 0002B0 | SRENSLCT DS | 22CL1 | SELECT BYTES |
| 0002C6 | SRENSRSP DS | CL32 | RESPONSE NAME |
| 0002E6 | SRENSREC DS | CL32 | RECORD NAME |
| 000306 | SRENVER DS | H | SELECT RECORD VERSION |
| 000308 | SRENPKS DS | CL2 | PF KEY SIMULATOR FIELD |
| 00030A | DS | CL12 | RESERVED FOR FUTURE USE |
| 000316 | SRENUNXT DS | CL8 | UMBRELLA NEXT TASK CODE |
| 00031E | SRENUNXF DS | CL1 | UMBRELLA NEXT TASK FLAG (0/N) |
| 00031F | DS | CL13 | RESERVED FOR FUTURE USE |
| 00032C | DS | CL8 | RESERVED FOR FUTURE USE |
| 001C0 | SRENPRT2 EQU | *-SRENMESS | LEN OF MOVE FROM SRO *JMA85064* |
| 00334 | SRENLEN EQU | ((*-SRE+3)/4)*4 | LENGTH OF NEW COPY IN BYTES |
| 000CD | SREPLENF EQU | ((*-SRE+3)/4) | LENGTH OF NEW COPY IN WORDS |
| 00034 | SREDO ORG | SREDG | DIALOG OPTIONS - TYPE 05 |
| 000034 | SREORVER DS | H | SELECT RECORD VERSION |
| 000036 | SREOCMOV DS | CL1 | COBOL MOVE (Y/N) |
| 000037 | SREOACTV DS | CL1 | ACTIVITY LOGGING (Y/N) |
| 000038 | SREOSYMB DS | CL1 | BUILD SYMBOL TABLE (Y/N) |
| 000039 | SREOMMDE DS | CL1 | MAP PAGE MODE (W/N/R) |
| 00003A | SREOMBKP DS | CL1 | MAP PAGE BACKPAGE OPTION(Y/N) |
| 00003B | SREOMUPD DS | CL1 | MAP PAGE UPDATE/BROWSE (U/B) |
| 00003C | SREOMADS DS | CL1 | MAP PAGE AUTO DISPLAY DKJ86210 |
| 00003D | SREOMESS DS | 4CL79 | MESSAGE AREAS |
| 000179 | SREOSLCT DS | 22CL1 | SELECT BYTES |
| 00018F | SREOPFKS DS | CL2 | PF KEY SIMULATOR |
| 000191 | SREOSRSP DS | CL32 | SELECT RESPONSE NAME |
| 0001B1 | SREOSREC DS | CL32 | SELECT RECORD NAME |
| 0001D1 | SREONTBL DS | CL1 | NO DIAGNOSTIC TABLES *LCB84199* |
| 0001D2 | SREONRLK DS | CL1 | NO RETRIEVAL LOCK *LMA86167* |

| | | | | |
|--------|----------|-----|-----------------|----------------------------------|
| 0001D3 | SREOPRFX | DS | CL2 | MESSAGE PREFIX *LMA86167* |
| 0001D5 | | DS | CL8 | RESERVED FOR FUTURE U**LMA86167* |
| 0001DD | SREOUNXT | DS | CL8 | UMBRELLA NEXT TASK CODE |
| 0001E5 | SREOUNXF | DS | CL1 | UMBRELLA NEXT TASK FLAG |
| 001E6 | SREOPRT1 | EQU | *-SRE | LEN OF MOVE FROM SRO *JMA85064* |
| 0001E6 | SREOENT | DS | CL1 | ENTRY POINT INDICATOR*JMA85064** |
| 0001E7 | | DS | C | RESERVED FOR FUTURE U*JMA85064* |
| 0001E8 | SREOIDD | DS | CL8 | INPUT MAP DDNAME |
| 0001F0 | SREOIDD | DS | CL8 | OUTPUT MAP DDNAME |
| 0001F8 | SREOSDD | DS | CL8 | SUSPENSE FILE DDNAME |
| 000200 | | DS | CL8 | RESERVED FOR FUTURE USE |
| 00208 | SREOLEN | EQU | ((*-SRE+3)/4)*4 | LEN OF DIALOG OPTION IN BYTES |
| 00082 | SREOLENF | EQU | ((*-SRE+3)/4) | LEN OF DIALOG OPTION IN WORDS |

1.122 #SRTDS

```

COPY #SRTDS
*****
***          SRT : ADSO SCREEN RECORD TABLE
***          ***
***          *** SRT IS A DSECT THAT DEFINES THE SCREEN RECORD TABLE USED
***          BY THE ADS ONLINE GENERATOR. THIS TABLE CONTAINS THE
***          ADDRESSES OF THE SRE'S (SCREEN RECORD ELEMENTS) FOR THE
***          DIALOG, PREMAP PROCESS, FIRST RESPONSE PROCESS, AND FIRST
***          NEW COPY RECORD SCREENS.
*****

```

Offset Value

| | | | |
|--------|-----------|---------------------|------------------------------------|
| 000000 | SRT | DSECT | 11:43:56 10/12/83 |
| 000000 | SRTID | DS CL4 | 'SRT*' |
| 000004 | SRTDSREA | DS A | ADDRESS OF DIALOG SRE |
| 000008 | SRTPSREA | DS A | ADDRESS OF PREMAP SRE |
| 00000C | SRTFRSEA | DS A | ADDR OF FIRST RESPONSE SRE |
| 000010 | SRTLRLSEA | DS A | ADDR OF LAST RESPONSE SRE |
| 000014 | SRTFNCPA | DS A | ADDR OF FIRST NEW COPY SRE |
| 000018 | SRTLNCPA | DS A | ADDR OF LAST NEW COPY SRE |
| 00001C | SRTCRSEA | DS A | ADDR OF CURRENT RESPONSE SCREEN |
| 000020 | SRTCNCPA | DS A | ADDR OF CURRENT NEW COPY SCREEN |
| 000024 | SRTNRSE | DS H | NUMBER OF RESPONSE SCREENS |
| 000026 | SRTNNCPY | DS H | NUMBER OF NEW COPY REC SCREENS |
| 000028 | SRTOLEN | EQU ((*-SRT+3)/4)*4 | LENGTH OF REL 1.1 AND EARLIER SRTS |
| 000028 | SRTOSREA | DS A | ADDRESS OF DIALOG OPTIONS SRE |
| 00002C | SRTTRLSE | DS H | RELEASE # OF GENERATOR SOFTWARE |
| 00002E | | DS H | RESERVED FOR FUTURE USE |
| 000030 | | DS 3F | RESERVED FOR FUTURE USE |
| 0003C | SRTLEN | EQU ((*-SRT+3)/4)*4 | LENGTH OF SRT IN BYTES |
| 0000F | SRTLENF | EQU ((*-SRT+3)/4) | LENGTH OF SRT IN WORDS |

1.123 #SSEDS

```
COPY #SSEDS
*****
***          SSE:    SUBSCRIPT STACK ENTRY
***          ***
***          *** SSE IS A DSECT THAT DESCRIBES AN ENTRY ON AN ADSO
***          *** SUBSCRIPT STACK. THIS STACK IS USED FOR RUNTIME
***          *** EVALUATION OF THE LOCATION OF A SUBSCRIPTED ITEM
***          *** WHERE THE SUBSCRIPTS MAY BE CONSTANTS, VARIABLES, OR
***          *** EXPRESSIONS. A SUBSCRIPT MAY ITSELF BE A SUBSCRIPTED
***          *** VARIABLE OR AN EXPRESSION CONTAINING A SUBSCRIPTED
***          *** VARIABLE. THERE IS NO LIMIT ON THE DEPTH OF NESTING.
***          ***
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------|----------------------------------|
| 000000 | SSE DSECT | 12:08:34 08/13/80 |
| 000000 | SSESBEG DS A | BEGINNING OF SUBS IN EVAL STACK |
| 000004 | SSECUROF DS F | CURRENT EVALUATED OFFSET TO ITEM |
| 000008 | SSESSNUM DS H | NUMBER OF SUBSCRIPTS EVALUATED |
| 00000A | SSESSEXP DS H | NUMBER OF SUBSCRIPTS EXPECTED |
| 00000C | SSEPRDVA DS A | ADDR OF PREVIOUS DOPE VECTOR |
| 000010 | SSEXPRA DS A | ADDRESS OF EXE |
| 00014 | SSELEN EQU *--SSE | LENGTH OF SSE ENTRY |

1.124 #STHDS

```

COPY #STHDS
*****
*** STHDS IS THE DSECT FOR THE SYMBOL TABLE HEADER. THERE IS ***
*** ONE HEADER FOR EACH COBOL, ASSEMBLER, AND PLI PROGRAM. THERE ***
*** IS ONE HEADER FOR EACH PROCESS WITHIN AN ADS/ONLINE DIALOG. ***
*** IF THERE IS MORE THAN ONE PROCESS IN AN ADSO DIALOG, THEN ***
*** THE HEADER OF THE FIRST PROCESS WILL CONTAIN THE ADDRESS OF ***
*** THE HEADER OF THE NEXT PROCESS, ETC. ***
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|--------------|---|
| 000000 | STHDS | DSECT | 08:42:16 05/30/84 |
| 000000 | STHNAME | DS CL8 | NAME OF PROGRAM/DIALOG |
| 000008 | STHPRNAM | DS CL32 | NAME OF PROCESS |
| 000028 | STHNPROA | DS A | A(NEXT PROCESS HDR IN THE CHAIN) |
| 00002C | STHSYMA | DS A | A(SYMTAB) |
| 000030 | STHSMDA | DS A | A(SYMDUP) |
| 000034 | STHDPPVA | DS A | A(DOPE VECTOR POOL) |
| 000038 | STHVALA | DS A | A(CONDITION VALUE TABLE) |
| 00003C | STHMILA | DS A | A(MAP INDEX LIST POOL) |
| 000040 | STHQALA | DS A | A(QALTAB) |
| 000044 | STHBLO | DS F | OFFSET TO BL CELLS IN COBOL PROG |
| 000048 | STHBLLO | DS F | OFFSET TO BLL CELLS IN COBOL PROG |
| 00004C | STHDXO | DS F | OFFSET TO INDEX CELLS IN COBOL PRG |
| 000050 | STHLABA | DS A | A(LABTAB) |
| 000054 | STHLBDA | DS A | A(LABDUP) |
| 000058 | STHLINA | DS A | A(LINTAB) |
| 00005C | STHLDPA | DS A | A(LINDUP) |
| 000060 | STHLNIDX | DS A | A(INDEX TO LINTABS FOR PL1) |
| 000064 | | DS 6F | RESERVED |
| 0007C | STHDSLEN | EQU *-STHDS | LENGTH OF PROCESS/PROGRAM TBL HDR |
| | | | ***** |
| | | | *** SYMHDR IS THE DSECT FOR THE HEADER FOR THE SYMTAB TABLE *** |
| | | | ***** |
| 000000 | SYMHDR | DSECT | SYMTAB HEADER |
| 000000 | SYMN | DS H | NUMBER OF ENTRIES IN TABLE |
| 000002 | SYMELEN | DS H | LENGTH IN BYTES OF LARGEST ENTRY |
| 000004 | SYMKEYOF | DS H | KEY OFFSET |
| 000006 | SYMKEYL | DS X | KEY LENGTH |
| | SYMBIN | #FLAG X'20' | DO A BINARY SEARCH |
| 000007 | SYMBINI | DS 0XL1 | |
| | 00020 | SYMBINM | EQU X'20' |
| 000007 | SYMFLAG | DS X | FLAG BYTE |
| 000008 | SYMSTART | DS A | POINTER TO START OF TABLE |
| | SYMHDLLEN | EQU *-SYMHDR | LENGTH OF HEADER |
| | | | ***** |
| | | | *** SYMDS IS A DSECT COMMON TO BOTH THE SYMTAB AND SYMDUP *** |
| | | | *** TABLES. THE SYMDUP FLAG INDICATES TO WHICH TABLE THE *** |
| | | | *** ENTRY BELONGS. *** |
| | | | ***** |
| 000000 | SYMDS | DSECT | SYMBOL HEADER |
| 000000 | SYMOFF | DS F | OFFSET OF SYMBOL INTO RECORD |
| * | | | OR OFFSET INTO SYMDUP FOR DUPS |
| 000004 | SYMOFF2 | DS F | FOR PL1, OFFSET INTO STATIC INT. |
| * | | | STORAGE FOR THE ADCON THAT IS |
| * | | | CREATED FOR STATIC EXTERNALS OR |
| * | | | OFFSET IN LABTAB TO ENTRY FOR |
| * | | | PROCEDURE THAT OWNS THIS SYMBOL |
| 000008 | | DS F | RESERVED |
| 00000C | SYMVALOF | DS H | OFFSET TO 1ST CONDITION VALUE |

| | | | |
|--------|--------------------------|------|------------------------------------|
| | * | | TABLE ENTRY |
| 00000E | SYMDVROF DS | H | OFFSET INTO DOPE VECTOR POOL |
| | * | | FOR OCCURRING FIELDS |
| 000010 | SYMMFINP DS | H | MAP FIELD INDEX |
| 000012 | SYMLEN DS | H | LENGTH OF SYMBOL |
| 000014 | SYMOCC DS | H | NUMBER OF OCCURRENCES |
| 000016 | SYMCLVL DS | H | OCCURS LEVEL |
| 000018 | SYMDECNM DS | H | NUMBER OF DECIMAL POSITIONS |
| 00001A | SYMQALOF DS | H | OFFSET TO ENTRY IN QALTAB FOR |
| | * | | QUALIFIER |
| 00001C | SYMMILOF DS | H | OFFSET TO MIL (FOR MAP FIELDS) |
| | * | | INTO MIL POOL |
| 00001E | SYMCELL DS | H | TABLE INDEX FOR BL(L) OR INDEX |
| | * | | CELL, OR ADS RECORD |
| 000020 | SYMBOFF DS | H | BIT OFFSET |
| 000022 | SYMPROC# DS | X | FOR PL1, PROCEDURE OWNING THIS |
| | * | | SYMBOL |
| 000023 | SYMDATYP DS | X | DATA TYPE FOR RHDCVAL |
| | SYMDUP #FLAG X'80' | | THIS ENTRY IS A SYMDUP ENTRY |
| 000024 | SYMDUPI DS | 0XL1 | |
| | 00080 SYMDUPM EQU X'80' | | ENTRIES IN VAL TABLE FOR THIS ELMT |
| 000024 | SYMCNDI DS | 0XL1 | |
| | 00040 SYMCNDM EQU X'40' | | ELEMENT IS MAP FIELD |
| 000024 | SYMMFLD #FLAG X'20' | | |
| | 00020 SYMMFLDM DS | 0XL1 | |
| | SYMMFLDM EQU X'20' | | |
| | SYMMIL #FLAG X'10' | | ON IF MIL IS IN MIL POOL |
| 000024 | SYMMILI DS | 0XL1 | |
| | 00010 SYMMILM EQU X'10' | | |
| 000024 | SYMFLAG1 DS | X | FIRST FLAG BYTE |
| | SYMADS #FLAG X'80' | | SYMCELL IS FOR ADS RECORD |
| 000025 | SYMADSI DS | 0XL1 | |
| | 00080 SYMADM EQU X'80' | | |
| | SYMBL #FLAG X'40' | | |
| 000025 | SYMBLI DS | 0XL1 | SYMCELL IS FOR BL CELL |
| | 00040 SYMBLM EQU X'40' | | |
| | SYMBLL #FLAG X'20' | | SYMCELL IS FOR BLL CELL |
| 000025 | SYMBLLI DS | 0XL1 | |
| | 00020 SYMBLLM EQU X'20' | | |
| | SYMIDX #FLAG X'10' | | SYMCELL IS FOR INDEX CELL |
| 000025 | SYMIDXI DS | 0XL1 | |
| | 00010 SYMIDXM EQU X'10' | | |
| 000025 | SYMFLAG2 DS | X | SECOND FLAG BYTE |
| | SYMAUTO #FLAG X'80' | | FIELD TYPE IS AUTOMATIC |
| 000026 | SYMAUTOI DS | 0XL1 | |
| | 00080 SYMAUTOM EQU X'80' | | |
| | SYMSTCI #FLAG X'40' | | FIELD TYPE IS STATIC INTERNAL |
| 000026 | SYMSTCII DS | 0XL1 | |
| | 00040 SYMSTCIM EQU X'40' | | |
| | SYMSTCX #FLAG X'20' | | FIELD TYPE IS STATIC EXTERNAL |
| 000026 | SYMSTCXI DS | 0XL1 | |
| | 00020 SYMSTCXM EQU X'20' | | |
| | SYMCONT #FLAG X'10' | | FIELD TYPE IS CONTROLLED |
| 000026 | SYMCONTI DS | 0XL1 | |
| | 00010 SYMCONTM EQU X'10' | | |
| | SYMBASE #FLAG X'08' | | FIELD TYPE IS BASED |
| 000026 | SYMBASEI DS | 0XL1 | |
| | 00008 SYMBASEM EQU X'08' | | |
| 000026 | SYMFLAG3 DS | X | THIRD FLAG BYTE |
| 000027 | SYNUMQ DS | X | NUMBER OF QUALIFIERS(SYMDUP ENTRY) |
| 000028 | ORG SYNUMQ | | |
| 000027 | SYMDCNT DS | X | DUPLICATION COUNT (SYMTAB ENTRY) |
| 000028 | SYMDPLEN EQU *-SYMDS | | LENGTH OF SYMDUP TABLE ENTRY |
| 000028 | SYMNAMA DS | CL8 | SYMBOL NAME FOR ASSEMBLER |

| | | | | | |
|---|-----------|-----------|---------|------------------------------------|------------------------------------|
| 000030 | SYMDSLNA | EQU | *-SYMDS | LENGTH OF ASSEMBLER SYMTAB | |
| | | ORG | SYMNAME | | |
| 000028 | SYMNAMEC | DS | CL32 | SYMBOL NAME FOR ADS, COBOL, PL1 | |
| | SYMDSLNC | EQU | *-SYMDS | LENGTH OF COBOL SYMTAB | |
| ***** | | | | | |
| *** DPVHDS AND DPVDS ARE THE DSECTS FOR THE DOPE VECTOR POOL *** | | | | | |
| *** IN THE SYMBOL TABLE, DPVHDS REPRESENTS THE DOPE VECTOR HEADER.*** | | | | | |
| *** DPVDS REPRESENTS A SINGLE DIMENSION WITHIN A DOPE VECTOR. *** | | | | | |
| *** THERE IS ONE DPVDS PER DIMENSION. *** | | | | | |
| ***** | | | | | |
| 000000 | DPVHDS | DSECT | | DOPE VECTOR HEADER | |
| 000000 | DPVHDOA | DS | F | DEPENDING ON CONTROL FIELD ADDR | |
| 000004 | | ORG | DPVHDOA | | |
| 000000 | DPVHCELL | DS | H | SYMCCELL # OF CONTROL FIELD | |
| 000002 | DPVHFST | DS | H | OFFSET WITHIN REC TO CONTROL FLD | |
| 000004 | DPVHNMD | DS | H | NUMBER OF DIMENSIONS IN ARRAY | |
| | DPVHFDF | #FLAG | X'80' | DEPENDING ON CTRL FIELD IS FULLWD | |
| 000006 | DPVHFDFI | DS | 0XL1 | | |
| | 000080 | DPVHFDFM | EQU | X'80' | |
| | | DPVHFDH | #FLAG | X'40' | DEPENDING ON CTRL FIELD IS HALFWD |
| 000006 | DPVHFDDHI | DS | 0XL1 | | |
| | 000040 | DPVHFDDHM | EQU | X'40' | |
| 000006 | DPVHFGLG | DS | X | FLAG BYTE | |
| 000007 | | DS | X | UNUSED | |
| | 000008 | DPVHLEN | EQU | *-DPVHDS | LENGTH OF DOPE VECTOR HEADER |
| 000000 | DPVDS | DSECT | | ONE PER DIMENSION | |
| 000000 | DPVDOFF | DS | H | FIELD OFFSET WITHIN CONTAINING OCC | |
| 000002 | DPVDSIZ | DS | H | SIZE OF A DIMENSION OCCURRENCE | |
| 000004 | DPVDMAX | DS | H | MAXIMUM SUBSCRIPT VALUE FOR DMNSN | |
| | 000006 | DPVDLEN | EQU | *-DPVDS | LENGTH OF ONE DIMENSION DESCRIPTOR |
| ***** | | | | | |
| *** VLHDS IS THE DSECT THAT DESCRIBES THE SYMVAL TABLE. THERE IS *** | | | | | |
| *** AT LEAST ONE VLH HEADER FOR EVERY SYMTAB ENTRY WHICH IS A *** | | | | | |
| *** CONDITION NAME. THE VLH HEADER IS FOLLOWED BY ONE OR TWO *** | | | | | |
| *** VALUES DEPENDING ON WHETHER THE VALUE IS A SINGLE VALUE OR A *** | | | | | |
| *** RANGE OF VALUES. *** | | | | | |
| ***** | | | | | |
| 000000 | VLHDS | DSECT | | VALUE HEADER | |
| 000000 | VLHNXTOF | DS | H | OFFSET OF NEXT ENTRY IN SYMVAL | |
| | * | | | FOR THIS CONDITION NAME | |
| 000002 | VLHVALEN | DS | H | LENGTH OF VALUE(S) | |
| | VLH2VAL | #FLAG | X'80' | THERE ARE 2 VALUES IN THIS ENTRY | |
| 000004 | VLH2VALI | DS | 0XL1 | | |
| | 000080 | VLH2VALM | EQU | X'80' | |
| 000004 | VLHFLAG | DS | X | FLAG BYTE | |
| 000005 | | DS | X | RESERVED | |
| 000006 | VLHVALS | DS | 0H | VALUE(S) START HERE | |
| | 000006 | VLHDSLEN | EQU | *-VLHDS | LENGTH OF THE VALUE HEADER |
| ***** | | | | | |
| *** MILDS IS THE DSECT THAT DESCRIBES THE SYMMIL TABLE. THERE IS *** | | | | | |
| *** ONE MIL HEADER FOR EVERY SYMTAB ENTRY WHICH IS A MULTIPLY *** | | | | | |
| *** OCCURRING FIELD AND IF ONE OF THOSE OCCURRENCES IS A MAPFIELD.*** | | | | | |
| *** EACH MIL HEADER GIVES THE NUMBER OF OCCURRENCES OF THE *** | | | | | |
| *** FIELD AND IS FOLLOWED BY A LIST OF HALFWORDS, ONE HALFWORD *** | | | | | |
| *** PER OCCURRENCE. *** | | | | | |
| ***** | | | | | |
| 000000 | MILDS | DSECT | | MAP INDEX LIST -- HEADER PORTION | |
| 000000 | MILNUM | DS | H | NUMBER OF INDICES FOR THIS FIELD | |
| 000002 | MILINDEX | DS | 0H | START OF FIRST INDEX (MAP INDEX | |
| | * | | | FOR THE ELEMENT OR -1) | |
| 000002 | MILDSLEN | EQU | *-MILDS | LENGTH OF INDICES FOR THIS FIELD | |
| ***** | | | | | |
| *** LABHDR IS THE HEADER FOR THE LABEL TABLE. *** | | | | | |
| *** LABDS IS A DSECT COMMON TO THE LABTAB AND LABDUP TABLES. *** | | | | | |
| *** FLAG SETTINGS INDICATE TO WHICH TABLE THE ENTRY BELONGS. *** | | | | | |

```

*****
000000 LABHDR DSECT          LABTAB HEADER
000000 LABN   DS H           NUMBER OF ENTRIES IN TABLE
000002 LABLEN DS H           LENGTH IN BYTES OF LARGEST ENTRY
000004 LABKEYOF DS H          KEY OFFSET
000006 LABKEYL DS X           KEY LENGTH
000007 LABFLAG DS X           FLAG BYTE
000008 LABSTART DS A          POINTER TO START OF TABLE
00000C LABHDLN EQU *-LABHDR
000000 LABDS  DSECT          LABEL TABLE ENTRY
000000 LABOFF DS F           OFFSET OF LABEL INTO PROGRAM OR
*                                         ENTRY IN LABDUP FOR DUPLICATES
000004 LABQUAL DS H           KEY LENGTH
000006 LABDUP #FLAG X'80'      LENGTH IN BYTES OF LARGEST ENTRY
000006 LABDUPI DS 0XL1        THIS IS A LABDUP ENTRY
000006 LABDUPM EQU X'80'      THIS IS A COBOL SECTION NAME
000006 LABSECI DS 0XL1        THIS IS AN ADSO SUBROUTINE NAME
000006 LABSECM EQU X'40'      THIS IS A PL1 PROCEDURE NAME
000006 LABSUB  DS 0XL1        THIS IS A PL1 PROCEDURE NAME
000006 LABSUBM EQU X'20'      THIS IS A PL1 PROCEDURE NAME
000006 LABPROC #FLAG X'10'      THIS IS A PL1 PROCEDURE NAME
000006 LABPROCI DS 0XL1       THIS IS A PL1 PROCEDURE NAME
000010 LABPROCM EQU X'10'      THIS IS A PL1 PROCEDURE NAME
000006 LABNOQL #FLAG X'08'      QUALIFIER NOT REQUIRED
000006 LABNQLI DS 0XL1        THIS IS A PL1 PROCEDURE NAME
000008 LABNQLM EQU X'08'      THIS IS A PL1 PROCEDURE NAME
000006 LABFLAG1 DS X           FLAG BYTE
000007 LABOCC# DS X           OCCURRENCE NUMBER OF INCLUDED
*                                         MODULE -- LABDUP ENTRY
000008 LABLAB  DS H           OFFSET TO LABTAB ENTRY FOR SECTION
*                                         THIS DUPLICATE PARAGRAPH BELONGS
*                                         IN (COBOL) -- LABDUP ENTRY
*                                         RESERVED -- LABDUP ENTRY
00000A DS H
00000C ORG LABOCC#
000007 LABDCNT DS X           LABEL DUPLICATION COUNT (# ENTRIES
*                                         IN LABDUP) -- LABTAB ENTRY
000008 LABPRBEG DS H           BEGINNING OFFSET OF PL1 PROCEDURE
*                                         -- LABTAB ENTRY
00000A LABPREDN DS H           ENDING OFFSET OF PL1 PROCEDURE
*                                         -- LABTAB ENTRY
00000C LABDUPLN EQU *-LABDS  LENGTH OF LABDUP ENTRY
00000C LABNAMAD DS CL8         ADS/ONLINE SUBROUTINE NAME
000014 LABDSLNA EQU *-LABDS  LENGTH OF LABTAB ENTRY FOR ADSO
00000C ORG LABNAMAD
00000C LABNAMC DS CL32        COBOL OR PL1 LABEL NAME
00000C LABDSLNC EQU *-LABDS  LENGTH OF LABTAB ENTRY FOR COBOL
*                                         AND PL1
*****
*** QALDS IS THE DSECT FOR THE QUALIFIER TABLE ***
*****
000000 QALDS  DSECT          QALTAB ENTRY
000000 QALNAME DS CL32        NAME OF QUALIFIER
000020 QALDSLEN EQU *-QALDS  QALTAB ENTRY LENGTH
*****
*** LINE NUMBER TABLE ***
*** LINHDR REPRESENTS THE LINTAB HEADER. ***
*** LINDS IS A DSECT COMMON TO THE LINTAB AND LINDUP TABLES. ***
*** FLAG SETTINGS INDICATE TO WHICH TABLE THE ENTRY BELONGS. ***
*****
000000 LINHDR DSECT          LINTAB HEADER
000000 LINN   DS H           NUMBER OF ENTRIES IN LINTAB TABLE
000002 LINELEN DS H          LENGTH IN BYTES OF LARGEST ENTRY
000004 LINKEYOF DS H          KEY OFFSET
000006 LINKEYL DS X           KEY LENGTH

```

| | | | |
|--------|----------|--------------|--|
| | LINBIN | #FLAG X'20' | DO A BINARY SEARCH |
| 000007 | LINBINI | DS 0XL1 | |
| 00020 | LINBINM | EQU X'20' | |
| 000007 | LINFLAG | DS X | FLAG BYTE |
| 000008 | LINSTART | DS A | POINTER TO START OF TABLE |
| 00000C | LINHDLEN | EQU *-LINHDR | |
| 000000 | LINDS | DSECT | LINE TABLE ENTRY |
| 000000 | LINOFF | DS F | OFFSET OF LINE INTO PROGRAM OR ENTRY IN LINDUP FOR DUPLICATES |
| * | | | RESERVED |
| 000004 | | DS F | |
| 000008 | LINQUAL | DS H | OFFSET TO MODULE NAME IN QALTAB |
| 00000A | LINIC | DS H | CODE FOR INSTRUCTION |
| 00000C | | DS CL2 | RESERVED |
| | LINDUP | #FLAG X'80' | THIS ENTRY IS IN LINDUP |
| 00000E | LINDUPI | DS 0XL1 | |
| 00080 | LINDUPM | EQU X'80' | |
| | LINNOQL | #FLAG X'40' | QUALIFIER NOT REQUIRED |
| 00000E | LINNOQLI | DS 0XL1 | |
| 00040 | LINNOQLM | EQU X'40' | |
| 00000E | LINFLAG1 | DS X | FLAG BYTE |
| 00000F | LINOCC# | DS X | OCCURRENCE NUMBER OF INCLUDED MODULE -- LINDUP ENTRY |
| * | | | |
| 000010 | | ORG LINOCC# | |
| 00000F | LINDCNT | DS X | LINE DUPLICATION COUNT (# ENTRIES IN LINDUP) -- LINTAB ENTRY |
| * | | | |
| 00010 | LINDUPLN | EQU *-LINDS | LENGTH OF LINDUP ENTRY |
| 000010 | LINNUM | DS F | LINE NUMBER |
| 00014 | LINTABLN | EQU *-LINDS | LENGTH OF LINTAB ENTRY |

1.125 #SWRDS

```

COPY #SWRDS
*****
***      SWR: ADS SWITCH RECORD
***      ***
***      SWR IS THE DSECT THAT DEFINES THE ADS/ONLINE SWITCH RECORD. ***
***      EACH BYTE OF THIS RECORD REPRESENTS A SWITCH WHICH IS SET   ***
***      WHEN THE HIGH ORDER BIT IS ON AND IS UNSET WHEN THE HIGH    ***
***      ORDER BIT IS OFF. THESE SWITCHES MAY BE TESTED BY THE USER   ***
***      AS RESERVED WORDS. THE DIALOG GENERATOR SETS UP CALLS TO     ***
***      RHDCEVAL TO EVALUATE THE TRUE/FALSE CONDITIONS OF THE       ***
***      SWITCH SETTINGS BASED ON ENCOUNTERING THE RESERVED WORDS IN ***
***      THE SYNTAX. THE RUN TIME SYSTEM MAKES SURE THE SWITCHES      ***
***      ARE PROPERLY SET.                                         ***
***      ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------------|-------------------------------------|
| 000000 | SWR | DSECT |
| | | PRINT NOGEN |
| 000000 | SWRBWDOF EQU *-SWR | OFFSET TO BWD SWITCH BYTE |
| | SWRBWD #FLAG X'80' | ON IF PAGE BACKWARD IN EFFECT |
| 000000 | SWRBWDSW DS X | PAGE BACKWARD SWITCH |
| 000D3 | SWRBWDADV EQU C'L' | PAGE BACKWARD PSEUDO-AID VALUE |
| | * | |
| 000001 | SWREODOF EQU *-SWR | OFFSET TO EOD SWITCH BYTE |
| | SWREOD #FLAG X'80' | ON IF END OF DETAILS IN EFFECT |
| 000001 | SWREODSW DS X | END OF DETAILS SWITCH |
| | * | |
| 000002 | SWRFWDOF EQU *-SWR | OFFSET TO FWD SWITCH BYTE |
| | SWRFWD #FLAG X'80' | ON IF PAGE FORWARD IN EFFECT |
| 000002 | SWRFWDSW DS X | PAGE FORWARD SWITCH |
| 000D2 | SWRFWDADV EQU C'K' | PAGE FORWARD PSEUDO-AID VALUE |
| | * | |
| 000003 | SWRDTLOF EQU *-SWR | OFFSET TO DTL SWITCH BYTE |
| | SWRDTL #FLAG X'80' | ON IF DETAIL FOUND |
| 000003 | SWRDTLSW DS X | DETAIL FOUND SWITCH |
| | * | |
| 000004 | SWRHDROF EQU *-SWR | OFFSET TO HDR SWITCH BYTE |
| | SWRHDR #FLAG X'80' | ON IF HEADER PROCESSING IN EFFECT |
| 000004 | SWRHDRSW DS X | HEADER PROCESSING SWITCH |
| 000D4 | SWRHDRADV EQU C'M' | HEADER PROCESSING PSEUDO-AID VALUE |
| | * | |
| 000005 | SWRFPGOF EQU *-SWR | OFFSET TO FPG SWITCH BYTE |
| | SWRFPG #FLAG X'80' | ON IF FULL PAGE CONDITION IN EFFECT |
| 000005 | SWRFPGSW DS X | FULL PAGE SWITCH |
| | * | |
| 000006 | SWRDNFOF EQU *-SWR | OFFSET TO DNF SWITCH BYTE |
| | SWRDNF #FLAG X'80' | ON IF DETAIL NOT FOUND |
| 000006 | SWRDNFSW DS X | DETAIL NOT FOUND SWITCH |
| | * | |
| 000007 | SWRMDPOF EQU *-SWR | OFFSET TO MDP SWITCH BYTE |
| | SWRMDP #FLAG X'80' | ON IF MAXIMUM DETAILS PUT |
| 000007 | SWRMDPSW DS X | MAXIMUM DETAILS PUT SWITCH |
| | * | |
| 000008 | SWRBATOF EQU *-SWR | OFFSET BATCH SWITCH BYTE |
| | SWRBAT #FLAG X'80' | ON IF BATCH MODE |
| 000008 | SWRBATSW DS X | BATCH MODE SWITCH |
| | * | |
| 000009 | SWRONLOF EQU *-SWR | OFFSET ONLINE SWITCH BYTE |

| | | |
|--------|---------------------|--------------------------------------|
| 000009 | SWRONL #FLAG X'80' | ON IF ONLINE MODE |
| | SWRONLSW DS X | ONLINE MODE SWITCH |
| | * | |
| 0000A | SWREOFOF EQU *-SWR | OFFSET END-OF-FILE SWITCH BYTE |
| | SWREOF #FLAG X'80' | ON IF END-OF-FILE |
| 0000A | SWREOFSW DS X | END-OF-FILE SWITCH |
| 000D5 | SWREOFAV EQU C'N' | END-OF-FILE PSEUDO-AID VALUE |
| | * | |
| 0000B | SWRIOEOF EQU *-SWR | OFFSET IOERROR SWITCH BYTE |
| | SWRIOE #FLAG X'80' | ON IF IO ERROR |
| 0000B | SWRIOESW DS X | IO ERROR SWITCH |
| 000D6 | SWRIOEAV EQU C'O' | IO-ERROR PSEUDO-AID VALUE |
| | * | |
| 0000C | SWRPGR OF EQU *-SWR | OFFSET TO PGR SWITCH BYTE |
| | SWRPGR #FLAG X'80' | ON IF PAGE READY CONDITION IN EFFECT |
| 0000C | SWRPGRSW DS X | PAGE READY SWITCH |
| | * | |
| | PRINT GEN | |
| 0000D | SWRDSLN EQU *-SWR | LENGTH OF SWITCH RECORD |

1.126 #SYADS

```

COPY #SYADS
*****
*** SYA -- WORK AREA FOR SYMBOL TABLE BUILDER. ***
*** ****
*****



Offset Value

000000      SYA      DSECT          16:23:29 11/01/83
000000      SYAID    DS   CL4       'SYA*' 
000004      SYAGREA  DS   F        SAVE CURRENT GRE ADDRESS
000008      SYAGRDA  DS   F        SAVE CURRENT GRD ADDRESS
00000C      SYAGPRA  DS   F        SAVE CURRENT GPR ADDRESS
000010      SYAQALA  DS   A        ADDRESS OF QALTAB STORAGE BLK
000014      SYADPVA  DS   A        ADDRESS OF SYMDPV STORAGE BLK
000018      SYAVALA  DS   A        ADDRESS OF SYMVAL STORAGE BLK
00001C      SYAMILA  DS   A        ADDRESS OF SYMMIL STORAGE BLK
000020      SYAISTA  DS   A        ADDRESS OF IST STORAGE BLK
000024      SYAILTA  DS   A        ADDRESS OF ILT STORAGE BLK
000028      SYASTA   DS   A        SYMBOL TABLE ADDRESS
* THE FOLLOWING ADDRESSES ARE ACTUALLY OFFSETS WITHIN THE
* SYMBOL TABLE LOAD MODULE
00002C      SYASYMA  DS   A        A(SYMTAB) -- OFFSET
000030      SYASMDA  DS   A        A(SYMDUP) -- OFFSET
000034      SYADOPA  DS   A        A(SYMDPV) -- OFFSET
000038      SYAVALUA DS   A        A(SYMLVAL) -- OFFSET
00003C      SYAMILIA DS   A        A(SYMMIL) -- OFFSET
000040      SYAQALTA DS   A        A(QALTAB) -- OFFSET
000044      SYALINEA DS   A        A(LINTAB) -- OFFSET
000048      SYALINDA DS   A        A(LINDUP) -- OFFSET
00004C      SYACURHA DS   A        A(CURRENT STH) -- OFFSET
000050      SYASTNAB DS   A        A(NEXT AVAIL BYTE) -- OFFSET
* END SYMBOL TABLE LOAD MODULE OFFSETS
000054      SYAMDCA  DS   A        ADDR OF FIRST MDC IN CHAIN
000058      SYASTEA  DS   A        A(END OF SYMBOL TBL) FOR ERR CHK
00005C      SYASYMEA DS   A        A(END OF SYMTAB) FOR ERROR CKING
000060      SYALNEEA DS   A        A(END OF LINTAB) FOR ERROR CKING
000064      SYAISTEA DS   A        A(END OF IST) FOR ERROR CHECKING
000068      SYAILTEA DS   A        A(END OF ILT) FOR ERROR CHECKING
00006C      SYAISTNA DS   F        POINTER TO NEXT AVAIL IST ENTRY
000070      SYAQALCT DS   F        COUNT OF QALTAB ENTRIES
000074      SYAISTCT DS   F        COUNT OF IST ENTRIES
000078      SYAILTCT DS   F        COUNT OF ILT ENTRIES
00007C      SYASDCT   DS   F        COUNT OF SYMDUP ENTRIES
000080      SYALNDCT DS   F        COUNT OF LINDUP ENTRIES
000084      SYADPVSZ DS   F        SIZE OF DOPE VECTOR POOL (SYMDPV)
000088      SYAMILSZ DS   F        SIZE OF MIL POOL (SYMDPV)
00008C      SYAVALSZ DS   F        SIZE OF VALUE POOL (SYMDPV)
000090      SYALINSZ DS   F        SIZE OF LINTABS (AGGREGATE)
000094      SYALDPSZ DS   F        SIZE OF LINDUPS (AGGREGATE)
000098      SYAQALSZ DS   F        SIZE OF QALTAB
00009C      SYAISTSZ DS   F        SIZE OF IST
0000A0      SYAILTSZ DS   F        SIZE OF ILT
0000A4      SYASMBSZ DS   F        SIZE OF SYMTAB
0000A8      SYASMDSZ DS   F        SIZE OF SYMDUP
0000AC      SYATOTSZ DS   F        SIZE OF SYMBOL TABLE - ALL PARTS
0000B0      SYAISTNO DS   F        NUMBER OF IST ENTRIES
0000B4      ISTBUF    DS   CL60    IST BUFFER
* FIELDS FOR THE SORT ROUTINE
0000F0      SYAPSTNA DS   A        A(1 BYTE PAST END OF TABLE)
0000F4      SYASTBLA DS   A        ADDRESS OF TABLE TO BE SORTED
0000F8      SYASRKYO DS   H        OFFSET WITHIN ENTRY TO SORT KEY

```

| | | | |
|--------|-----------|---------------------|------------------------------------|
| 0000FA | SYASRKYL | DS H | LENGTH OF SORT KEY |
| 0000FC | SYASENTL | DS H | LENGTH OF ONE ENTRY IN TABLE |
| 0000FE | TMPENTRY | DS CL60 | TEMPORARY ENTRY FOR SWAP |
| | SYASWAP | #FLAG X'80' | SWAP SWITCH FOR SORT |
| 00013A | SYASWAPI | DS OXL1 | |
| | 00080 | SYASWAPM EQU X'80' | |
| | | ADDED1 #FLAG X'40' | ALREADY ADDED 1 FOR 1ST DUP |
| 00013A | ADDED1I | DS OXL1 | |
| | 00040 | ADDED1M EQU X'40' | |
| 00013A | SYAFLAGS | DS X | FLAG BYTE |
| | * | END FIELDS FOR SORT | |
| 00013C | SYAILTNO | DS F | # ILT ENTRIES (TRUE #) |
| 000140 | SYALNCNT | DS F | COUNT OF LINE NUMBERS |
| 000144 | SYASVR7 | DS F | SAVE R7 |
| 000148 | SYASVR2 | DS F | SAVE R2 FOR HI LEVEL SUBROUTINES |
| 00014C | SYASVR2A | DS F | SAVE R2 FOR LINTAB SUBROUTINES |
| 000150 | SYASVR2B | DS F | SAVE R2 FOR SPECIST SUBROUTINE |
| 000154 | SYAMOD2 | DS F | SAVE R2 FOR DOMOD |
| 000158 | SYAGPC2 | DS F | SAVE R2 FOR GETPROC |
| 00015C | SYAGENR2 | DS F | SAVE R2 IN ADSOGENR |
| | * | | |
| 000160 | SYASVNAME | DS CL32 | SAVE NAME OF IST SYMBOL |
| 000180 | SYASVNUM | DS F | SAVE ILT LINE NUMBER |
| 000184 | SYACMXA | DS F | SAVE CME OFFSET FOR ILT |
| 000188 | SYARECCT | DS H | # ADS RECORDS (INC. VDB, ETC.) |
| 00018A | SYAPRCT | DS H | # PROCESSES IN THE DIALOG |
| 00018C | SYASMCT | DS H | COUNT OF SYMTAB ENTRIES |
| 00018E | SYALINCT | DS H | COUNT OF LINTAB ENTRIES |
| 000190 | SYAGLEOF | DS H | LITPOOL OFFSET OF GLE (DPV OR MIL) |
| 000192 | SYAPLOF | DS H | POOL OFFSET OF THIS ENTRY |
| 000194 | SYADMSZ | DS H | DOPE VECTOR OR MIL LENGTH |
| 000196 | SYAQLCUR | DS H | CURRENT SYMBOL'S QALTAB OFFSET |
| 000198 | SYAOCCNO | DS H | MAX # MIL OCCURRENCES |
| 00019A | SYAMLIN1 | DS H | MAX LINES IN A MODULE IN A PROCSS |
| 00019C | SYAMLIN2 | DS H | 2ND MAX LINES IN A MOD IN A PROC |
| 00019E | SYAMOCC1 | DS H | OCC. CT FOR MAX LINES IN A MOD |
| 0001A0 | SYARLDCT | DS H | COUNT OF RLDS PUT IN CURR SYMCTL |
| 0001A2 | SYARLTOT | DS H | TOTAL # RLDS FOR ALL SYMCTLs |
| 0001A4 | SYAMILOF | DS H | OFFSET WITHIN SYMMIL POOL |
| 0001A6 | SYAVALOF | DS H | OFFSET WITHIN SYMVAL POOL |
| 0001A8 | SYADPVOF | DS H | OFFSET WITHIN SYMDPV POOL |
| 0001AA | SYAQALOF | DS H | OFFSET WITHIN QALTAB |
| 0001AC | SYASYMCL | DS H | SYMCELL VALUE FOR CURRENT REC |
| 0001AE | SYADLGVR | DS H | DIALOG VERSION |
| 0001B0 | SYADLGNM | DS CL8 | DIALOG NAME |
| | SYADOCT | #FLAG X'80' | DO COUNT ONLY (DPV, MIL PROCSSING) |
| 0001B8 | SYADOCTI | DS OXL1 | |
| | 00080 | SYADOCTM EQU X'80' | |
| | | SYAPRM3 #FLAG X'40' | THERE IS A PARM IN R3 FOR WTL |
| 0001B8 | SYAPRM3I | DS OXL1 | |
| | 00040 | SYAPRM3M EQU X'40' | |
| | | NEWSTH #FLAG X'20' | BUILD NEW STH -NOT THE 1ST PROCESS |
| 0001B8 | NEWSTHI | DS OXL1 | |
| | 00020 | NEWSTHM EQU X'20' | |
| | | SYADUP #FLAG X'10' | PROCESS A DUPLICATE ENTRY |
| 0001B8 | SYADUPI | DS OXL1 | |
| | 00010 | SYADUPM EQU X'10' | |
| | | SYALMVC #FLAG X'08' | EXECUTED THE LAST MOVE TO SYMTEXT |
| 0001B8 | SYALMVCI | DS OXL1 | |
| | 00008 | SYALMVCM EQU X'08' | |
| | | STRTDON #FLAG X'04' | SYMSTART ENTRY ALREADY IN SYMCTL |
| 0001B8 | STRTDONI | DS OXL1 | |
| | 00004 | STRTDONNM EQU X'04' | |
| 0001B8 | SYAFLAG | DS X | FLAG BYTE |
| 001BC | SYAWKLEN | EQU ((*-SYA+3)/4)*4 | LENGTH OF SYAWRK IN BYTES |
| 0006F | SYAWKLNF | EQU ((*-SYA+3)/4) | LENGTH OF SYAWRK IN WORDS |

1.127 #TAEDS

COPY #TAEDS

```
*****
***          TAE:  ADS APPLICATON CONTROL FACILITY      ***
***          TASK APPLICATION TABLE ENTRY                 ***
***          ***
***          TAE IS A DSECT THAT DEFINES A TASK APPLICATION TABLE ENTRY ***
***          USED BY THE ADS APPLICATION CONTROL FACILITY. THERE IS      ***
***          A TAE FOR EACH TASK NAME WHICH MAY BE USED TO INITIATE AN      ***
***          APPLICATION. THE ENTRY CONTAINS THE TASK NAME AND THE      ***
***          APPLICATION NAME AND VERSION NUMBER WHICH IT INITIATES.    ***
***          ***
*****
```

Offset Value

| | | | |
|--------|----------|-----------|---------------------|
| 000000 | TAE | DSECT | 15:02:17 09/30/82 |
| 000000 | TAETNAME | DS CL8 | TASK NAME |
| 000008 | TAEANAME | DS CL8 | APPLICATION NAME |
| 000010 | TAEAVER | DS H | APPLICATION VERSION |
| 00012 | TAELEN | EQU *-TAE | LENGTH OF ONE TAE |

1.128 #TATUDS

COPY #TATUDS

Offset Value

| | | |
|--------|----------------------------|-------------------|
| 000000 | TATUREC1 DSECT | 11:13:17 09/11/85 |
| 000000 | ADSENTP DS 0CL12 | |
| 000000 | TFULWRD0 DS 3FL4 | |
| 00000C | ADSCSAA DS FL4 | |
| 000010 | ADSIDMSA DS FL4 | |
| 000014 | TCTAPA DS FL4 | |
| 000018 | TMSGCODE DS PL4 | |
| 00001C | TMSGIX DS FL4 | |
| 000020 | TR12EP1 DS FL4 | |
| 000024 | TR12SUB1 DS FL4 | |
| 000028 | TR13NTRY DS FL4 | |
| 00002C | TR14NTRY DS FL4 | |
| 000030 | TSIMGSAA DS FL4 | |
| 000034 | TSTGADDR DS FL4 | |
| 000038 | TSTGLEN DS FL4 | |
| 00003C | TSTKTATU DS 0CL400 | |
| 00003C | TFULWRD1 DS 100FL4 | |
| 0001CC | TSYSPSLST DS 0CL40 | |
| 0001CC | TFULWRD2 DS 10FL4 | |
| 0001F4 | TWORKA DS FL4 | |
| 0001F8 | TAPPVERS DS HL2'1' | |
| 0001FA | TDTAILIX DS HL2'0' | |
| 0001FC | TDTAILMX DS HL2'4' | |
| 0001FE | TFUNC DS HL2'0' | |
| 000200 | TTATVERS DS HL2'1' | |
| 000202 | TAPPACTN DS CL3'ADD' | |
| 000205 | TAPPLOC DS CL1'A' | |
| 000206 | TAPPNAME DS CL8 | |
| 00020E | TDICNODE DS CL8 | |
| 000216 | TDICDBN DS CL8 | |
| 00021E | TENVIRMT DS CL1'0' | |
| 00021F | TFLAGBGN DS CL1 | |
| 000220 | TRUBOUND DS CL1'N' | |
| 000221 | TTATLOC DS CL1'A' | |
| 000222 | TTATNEW DS CL1'N' | |
| 000223 | TTATNAME DS CL8'\$ACF@TAT' | |
| 00022B | TXTOTAT DS CL1 | |
| 00022C | TXSTG DS CL160 | |

1.129 #TFEDS

```
COPY #TFEDS
*****
***      TFE: ADS APPLICATION GENERATOR TASK/TOP FUNCTION
***      TFE IS THE DSECT THAT DEFINES THE INDEX OF THE TOP FUNCTION ***
***      TO BE INVOKED WHENEVER A GIVEN TASK CODE IS ENTERED FOR AN ***
***      APPLICATION.
***      ****
*****
```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|-----------|
| 000000 | TFE | DSECT |
| 000000 | TFETNAME | DS CL8 |
| 000008 | TFEFCNIX | DS H |
| 00000A | | DS H |
| 00000C | | DS 2F |
| 00014 | TFELEN | EQU *-TFE |

| | | | |
|--------|----------|-----------|--------------------------------|
| 000000 | TFE | DSECT | 18:16:45 03/01/83 |
| 000000 | TFETNAME | DS CL8 | TASK NAME |
| 000008 | TFEFCNIX | DS H | INDEX OF TOP FUNCTION FOR TASK |
| 00000A | | DS H | RESERVED |
| 00000C | | DS 2F | RESERVED |
| 00014 | TFELEN | EQU *-TFE | LENGTH OF TFE |

1.130 #URNDS

```

COPY #URNDS
*****
***     URN: ADS APPLICATION GENERATOR USING RECORD PARM (INTERNAL) ***
***     URN IS A DSECT THAT IS THE INTERNAL CONTROL BLOCK USED TO   ***
***     HOLD A RECORD NAME FOR A USING RECORD LIST FOR A FUNCTION.   ***
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|----------------------------|------------------------------|
| 000000 | URN DSECT | 12:24:04 04/04/85 |
| 000000 | URNNXTA DS A | ADDR OF NEXT URN |
| 000004 | URNPREVA DS A | ADDR OF PREVIOUS URN |
| 000008 | URNSEQN DS H | USING RECORD SEQUENCE NUMBER |
| 00000A | URNNAME DS CL32 | USING RECORD NAME |
| 0002A | URNLEN EQU *-URN | LENGTH OF URN |
| 0000B | URNLENF EQU ((*-URN+3)/4) | LENGTH OF URN IN FULL WORDS |

1.131 #VDBDS

```

COPY #VDBDS
*****
***          VDB: VARIABLE DIALOG BLOCK
***
***      VDB IS A DSECT THAT DESCRIBES THE VARIABLE DIALOG BLOCK
***      USED BY THE ADSO RUNTIME SYSTEM. THE VDB IS A
***      NON REENTRANT TABLE CONTAINING USER SPECIFIC INFORMATION
***      ABOUT A PARTICULAR DIALOG. A VDB IS DYNAMICALLY
***      CREATED FOR EACH USER OF A DIALOG WHEN THE DIALOG
***      IS INITIATED. THE VDB RESIDES IN MAIN MEMORY AND
***      IT CONTAINS HEADER INFORMATION FOLLOWED BY THE MRB FOR
***      THE DIALOG'S MAP (IF ANY) FOLLOWED BY A TABLE OF VRE'S
***      CONTAINING VARIABLE RUN TIME INFORMATION FOR EACH RECORD
***      NEEDED BY THIS DIALOG. SEE THE COMMENTARY PRECEEDING THE
***      VRE DESCRIPTION FOR INFORMATION ABOUT SPECIAL RECORDS
***      FOR WHICH VRE'S WILL BE CREATED.
***      MRB ADDRESS (ONLINE)
*****

```

| <u>Offset</u> | <u>Value</u> | |
|---------------|--------------|--|
| 000000 | VDB | DSECT |
| 000000 | VDBID | DS CL4 |
| 000004 | VDBDNAM | DS CL8 |
| 00000C | VDBPREVA | DS A |
| 000010 | VDBNEXTA | DS A |
| 000014 | VDBMRBA | DS A |
| 000018 | | ORG VDBMRBA |
| 000014 | VDBDITA | DS A |
| 000018 | VDBVRETA | DS A |
| 00001C | VDBIRAVX | DS H |
| 00001E | VDBSWRVX | DS H |
| 000020 | VDBCMEOF | DS F |
| 000024 | VDBPRCOF | DS F |
| 000028 | VDBLRPOF | DS F |
| 00002C | VDBRATOF | DS F |
| 000030 | VDBSRVLV | DS X |
| 000031 | | ORG VDBSRVLV |
| 000030 | VDBSRTN | DS 10F |
| 000058 | VDBNRECS | DS H |
| 00005A | VDBMRBSZ | DS H |
| 00005C | VDBMRBBL | DS H |
| 00005E | VDBMREL | DS H |
| | | ** FOLLOWING FIELDS ARE XDE ADDRESSABLE DATA ELEMENTS ** |
| 000060 | VDBCURPS | DS XL4 |
| 000064 | | ORG VDBCURPS |
| 000060 | VDBROWOF | EQU *-VDB |
| 000060 | VDBCURRW | DS H |
| 000062 | VDCOLOF | EQU *-VDB |
| 000062 | VDCURCL | DS H |
| 000064 | VDBDBOF | EQU *-VDB |
| 000064 | VDBERRST | DS CL4 |
| 000068 | VDBJDTOF | EQU *-VDB |
| 000068 | VDBJULDT | DS PL3 |
| 00006B | VDBGRGDT | DS PL4 |
| 00006F | VDBTIME | DS PL4 |
| 000073 | VDBFSTOF | EQU *-VDB |
| | | PRINT NOGEN |
| | VDBDFT | #FLAG X'80' |
| | VDBDINV | #FLAG X'40' |
| | VDBIIPM | #FLAG X'20' |
| | | ON = 1ST TIME DIALOG EXECUTION |
| | | ON = CURRENT CME IS LINK |
| | | ON = LINK ISSUED BY PREMAP PROCESS |

| | | |
|--------|---|--------------------------------------|
| | VDBDCLR #FLAG X'10' | ON = DIALOG CONTAINS LOGICAL RECS. |
| | VDBREDY #FLAG X'08' | ON = AREAS TO BE READIED |
| | VDBBNDF #FLAG X'04' | ON = DIALOG DID NOT BIND RUN UNIT |
| | VDBINNS #FLAG X'02' | ON = NOT SAVING CURRENCY *JEB84047* |
| | VDBRALL #FLAG X'01' | ON = READY ALL FOR PROC *JEB84151* |
| | PRINT GEN | |
| 000073 | VDBFLAG DS X | FLAG BYTE |
| 000074 | VDBPASOF EQU *-VDB | VDB OFFSET TO PATH STATUS RETURN |
| 000074 | VDBPATST DS CL16 | LATEST PATH STATUS FREOM LR CALL |
| 000084 | DS OF | |
| 000084 | VDBDDBOF EQU *-VDB | VDB OFFSET TO DIRECT DBKEY |
| 000084 | VDBDRDBK DS F | DIRECT DBKEY, LOCATION MODE DIRECT |
| 000088 | 00088 VDBGDTOF EQU *-VDB | VDB OFFSET TO GREGORIAN DATE |
| 000088 | VDBGREG DS CL6 | GREGORIAN DATE (UNPACKED YYMMDD) |
| 00008E | 0008E VDBTIMOF EQU *-VDB | VDB OFFSET TO TIME |
| 00008E | VDBTME DS CL6 | TIME (UNPACKED HHMSS) |
| 000094 | 00094 VDBNODOF EQU *-VDB | VDB OFFSET TO NODE-NAME |
| 000094 | VDBNODE DS CL8 | NODE-NAME |
| 00009C | 0009C VDBDBNOF EQU *-VDB | VDB OFFSET TO DB-NAME |
| 00009C | VDBDBN DS CL8 | DB-NAME |
| 0000A4 | 000A4 VDBUALN EQU *-VDB | END USER ADDRESSABLE FLDS*JEB84261* |
| 0000A4 | VDBFDEA DS F | CORRESPONDING FDE OFFSET FOR ACF |
| 0000A8 | 0000A8 VDBDTIME DS CL16 | DIALOG'S DATE/TIME STAMP |
| | PRINT NOGEN | |
| | VDBDSMP #FLAG X'80' | DIALOG STARTED MAP PAGING*JEB84261* |
| | VDBEXRU #FLAG X'40' | DIALOG PASSED DOWN HIS RU*JEB85165* |
| | VDBCUSV #FLAG X'20' | DIALOG SAVED CURRENCIES *PHH86006* |
| | VDBPAGE #FLAG X'10' | DIALOG SAVED \$PAGE FIELD ECM86013 |
| | VDBNRLK #FLAG X'08' | RETRIEVAL READY DONE ECM86013 |
| | *** | WITHOUT RETRIEVAL LOCKING ECM86160 |
| | VDBINUS #FLAG X'04' | ISSUED 'TRUST ME' CTL CMD*PHH86174*3 |
| | VDBTCLR #FLAG X'02' | ON = THREAD CONTAINS LR'S*PHH86177* |
| | VDBINUM #FLAG X'01' | ON = MRB IS IN A PREV VDB ECM86181 |
| | PRINT GEN | |
| 0000B8 | VDBFLAG2 DS X | FLAG BYTE *JEB84261* |
| | PRINT NOGEN | ECM86191 |
| | VDBLFMC #FLAG X'80' | DIALOG WAS LINK'ED FROM ECM86191 |
| | *** | USING MAP-CONTRL ECM86191 |
| | PRINT GEN | ECM86191 |
| 0000B9 | VDBFLAG3 DS X | FLAG BYTE ECM86191 |
| 0000BA | DS XL2 | RESERVED *JEECM86191 |
| 0000BC | 000BC VDBECTOF EQU *-VDB | VDB OFFSET TO ERROR COUNT |
| 0000BC | VDBECNT DS F | ERROR COUNT |
| 0000C0 | 000C0 VDBICTOF EQU *-VDB | VDB OFFSET TO INPUT COUNT |
| 0000C0 | VDBICNT DS F | INPUT COUNT |
| 0000C4 | 000C4 VDOCTOF EQU *-VDB | VDB OFFSET TO OUTPUT COUNT |
| 0000C4 | VDOBCT DS F | OUTPUT COUNT |
| 0000C8 | VDBSPAGE DS F | SAVE \$PAGE FIELD ECM86013 |
| 0000CC | VDBRALID DS F | REC ALLOC ID FOR FASTLINK*PHH86174*3 |
| 0000D0 | VDBVRETX DS F | ADDR OF RDE EXTENSIONS *PHH86180*3 |
| 0000D4 | 000D4 VDBJDTXO EQU *-VDB | OFFSET TO EXTENDED JULIAN JMA90178 |
| 0000D4 | VDBJULDX DS PL4 | JULIAN (YYYYDDD+) PACKED JMA90178 |
| 0000D8 | VDBGRGDX DS PL5 | GREGORIAN (YYYYMMDD+) PCKD JMA90178 |
| 0000D8 | 000DD VDBGDTXO EQU *-VDB | OFFSET TO EXTNDED GREGORIAN JMA90178 |
| 0000DD | VDBGREGX DS CL8 | GREGORIAN (UNPKD YYYYMMDD) JMA90178 |
| 0000E8 | VDBMRB DS OA | START OF MRB (IF ANY) |
| 0000E8 | 000E8 VDBLEN EQU *-VDB | LENGTH OF VDB FIXED PORTION |
| | ***** | ***** |
| | *** | *** |
| | *** VRE: VDB RECORD BUFFER ADDRESS ENTRY | *** |
| | *** | *** |
| | *** VRE IS A DSECT THAT DEFINES AN ENTRY IN THE VDB | *** |
| | *** CONTAINING INFORMATION ABOUT WHERE A RECORD BUFFER IS | *** |
| | *** LOCATED IN THE RBB FOR A RECORD USED BY THE DIALOG. | *** |
| | *** THERE IS A VRE FOR EACH RECORD USED BY THE DIALOG AND | *** |
| | *** THEY ARE IN THE SAME ORDER AS THE RDE'S IN THE FDB. | *** |

```

*** THE VRE'S APPEAR IN THE VDB IMMEDIATELY FOLLOWING THE ***
*** MRB. VRE'S MAY BE CREATED FOR THREE SPECIAL "RECORDS" ***
*** IF RDE'S EXIST FOR THEM. THESE SPECIAL VRE'S ARE USED TO ***
*** CONTAIN THE ADDRESSES OF THE BEGINNING OF THE VDB, THE ***
*** INTERMEDIATE RESULT AREA (IRA) FOR RHDCEVAL AND THE ***
*** DIALOG'S LITERAL POOL.
*** ****
***** ****
000000    VRE      DSECT          12:37:52 07/09/80
              VREALOC #FLAG X'80'          ON IF DIALOG ALLOCATED BUFF
000000    00080   VREALOCI DS 0XL1
              VREALOCM EQU X'80'
              VRERNIB #FLAG X'40'          ON IF RECORD NOT IN RBB
000000    00040   VRERNIBI DS 0XL1
              VRERNIBM EQU X'40'
              VRELRL #FLAG X'20'          ON IF RECORD IS A LOGICAL RECORD
000000    00020   VRELRLI DS 0XL1
              VRELRM EQU X'20'
              VREPSR #FLAG X'10'          ON IF RECORD IS A LR ELEMENT
000000    00010   VREPSRI DS 0XL1
              00010   VREPSRM EQU X'10'
000000    00000   VREFLAG DS XL1
              00001   DS XL3             RESERVED
000004    VRERCA  DS A             RECORD ADDRESS
000008    VRERBBIX DS X            INDEX OF RBB CONTAINING BUFF
000009    ORG     VRERBBIX
000008    VRERBEOF DS F            OFFSET TO RBE IN RBB FOR RECORD
00000C    VRESTOBT DS CL4          ERROR-STATUS LAST RETRIEVE DML CMD
000010    DS     0F              TO BE SURE VRE'S ARE WORD ALIGNED
              00010   VRELEN  EQU *-VRE LENGTH OF VRE ENTRY

```

1.132 #WKBCTL

```

COPY #WKBCTL
*****
* THIS DSECT MAPS THE WORK AREA FOR THE BATCH MAPPING CONTROLLER *
*****
```

| <u>Offset</u> | <u>Value</u> | | |
|---------------|--------------|---|----------------------------------|
| 000000 | WRKBCTL | DSECT | 08:35:05 05/28/87 |
| 000000 | WRKID | DS CL8 | EYECATCHER |
| 000008 | SYSPLIST | DS 10F | GENERAL PARAMETER LIST AREA |
| 000030 | WRKDBL | DS D | WORK DOUBLEWORD |
| 000038 | WRKTBLA | DS A | A(TABLE TO BE SEARCHED) |
| 00003C | WRKKEYA | DS A | A(SEARCH ARG) |
| 000040 | WRKKEYL | DS F | LENGTH OF KEY |
| 000044 | WRKBOFF | DS F | OFFSET TO KEY IN TABLE ENTRY |
| | * | | |
| 000048 | | ORG WRKKEYA | |
| 00003C | WRKHEXSA | DS A | A(CVRTHEX SOURCE STRING) |
| 000040 | WRKHEXTA | DS A | A(CVRTHEX TARGET STRING) |
| 000044 | WRKHEXLN | DS A | LENGTH OF CVRTHEX SOURCE STRING |
| | * | | |
| 000048 | WRKFMDEA | DS A | A(FMD TABLE ENTRY) |
| | * | | |
| 00004C | WRKMSGA | DS A | A(LEVEL 2 MSG ID) |
| 000050 | WRKFCNA | DS A | A(FUNCTION REQUESTED REPL PARM) |
| 000054 | WRKSVRC | DS A | RETURN CODE SAVE AREA |
| 000058 | WRKSVRA | DS (7-2+1)A | SAVE AREA FOR REGS 2-7 (CVRTHEX) |
| 000070 | WRKSVP1 | DS F | SAVE AREA FOR PARM ADDR |
| | * | | |
| 000074 | WRKBYTES | DS 0XL4 | WORK BYTES |
| 000074 | WRKBYTE0 | DS X | WORK BYTE 0 |
| 000075 | WRKBYTE1 | DS X | WORK BYTE 1 |
| 000076 | WRKBYTE2 | DS X | WORK BYTE 2 |
| 000077 | WRKBYTE3 | DS X | WORK BYTE 3 |
| | * | | |
| 000078 | WRKFLAGS | DS 0XL4 | BCTL INTERNAL FLAGS |
| | WRKBASE | #FLAG X'80' | BASE FILE REQUEST |
| 000078 | WRKBASEI | DS 0XL1 | |
| | 000080 | WRKBASEM EQU X'80' | |
| | WRKSUPP | #FLAG X'40' | SUPPLEMENTARY FILE REQUEST |
| 000078 | WRKSUPPI | DS 0XL1 | |
| | 000040 | WRKSUPPM EQU X'40' | |
| | WRKDEFN | #FLAG X'20' | DEFINITION ERROR |
| 000078 | WRKDEFNI | DS 0XL1 | |
| | 000020 | WRKDEFNM EQU X'20' | |
| | WRKPHYS | #FLAG X'10' | PHYSICAL ERROR |
| 000078 | WRKPHYSI | DS 0XL1 | |
| | 000010 | WRKPHYSM EQU X'10' | |
| | WRKLGCL | #FLAG X'08' | LOGICAL ERROR |
| 000078 | WRKLGCLI | DS 0XL1 | |
| | 000008 | WRKLGCLM EQU X'08' | |
| | WRKFRPL | #FLAG X'04' | FREE THE BRPL |
| 000078 | WRKFRPLI | DS 0XL1 | |
| | 000004 | WRKFRPLM EQU X'04' | |
| | WRKEALL | #FLAG WRKLGCLM+WRKPHYSM+WRKDEFNM | ALL ERROR TYPES |
| 000078 | WRKEALLI | DS 0XL1 | |
| | 000038 | WRKEALLM EQU WRKLGCLM+WRKPHYSM+WRKDEFNM | |
| 000078 | WRKFLAG0 | DS X | FLAG BYTE 0 |
| | WRKLZ | #FLAG X'80' | LRECL IS ZERO |
| 000079 | WRKLZI | DS 0XL1 | |
| | 000080 | WRKLZM EQU X'80' | |
| | WRKBZ | #FLAG X'40' | BLOCKSIZE IS ZERO |
| 000079 | WRKBZI | DS 0XL1 | |

| | | | | | |
|--------|-------|----------|-------|-----------------|-----------------------------------|
| | 00040 | WRKBZM | EQU | X'40' | |
| | | WRKLNZ | #FLAG | X'20' | LRECL IS NOT ZERO |
| 000079 | | WRKLNZI | DS | 0XL1 | |
| | 00020 | WRKLNZM | EQU | X'20' | BLOCKSIZE IS NOT ZERO |
| | | WRKBNZ | #FLAG | X'10' | |
| 000079 | | WRKBNZI | DS | 0XL1 | |
| | 00010 | WRKBNZM | EQU | X'10' | |
| 000079 | | WRKFLAG1 | DS | X | FLAG BYTE 1 |
| 00007A | | WRKUNIT | DS | X | UNIT USED |
| 00007B | | WRKRECFM | DS | X | RECORD FORMAT USED |
| 00007C | | WRKLRECL | DS | H | LOGICAL RECORD LENGTH USED |
| 00007E | | WRKBLKSZ | DS | H | BLOCK SIZE USED |
| 000080 | | WRKDEVNM | DS | F | DEVICE NAME USED |
| | | * | | | |
| 000084 | | WRKEDT1L | DS | X | LENGTH OF EDIT WORK AREA 1 |
| 000085 | | WRKEDT1 | DS | XL16 | EDIT WORK AREA 1 |
| 000095 | | WRKEDT2L | DS | X | LENGTH OF EDIT WORK AREA 2 |
| 000096 | | WRKEDT2 | DS | XL16 | EDIT WORK AREA 2 |
| 0000A6 | | WRKEDT3L | DS | X | LENGTH OF EDIT WORK AREA 3 |
| 0000A7 | | WRKEDT3 | DS | XL16 | EDIT WORK AREA 3 |
| 0000B7 | | WRKFNAML | DS | X | LENGTH OF FILE NAME |
| 0000B8 | | WRKFNAM | DS | CL8 | FILE NAME |
| 0000C0 | | WRKRCL | DS | X | LENGTH OF DISPLAYABLE RETURN CODE |
| 0000C1 | | WRKRC | DS | CL8 | DISPLAYABLE RETURN CODE |
| 0000C9 | | WRKDVCIL | DS | X | LENGTH OF DEVICE INFO |
| 0000CA | | WRKDVC | DS | CL6 | SYSNO (IF PRESENT) |
| 0000D0 | | WRKEQUAL | DS | X | EQUAL SIGN |
| 0000D1 | | WRKCUU | DS | CL3 | UNIT ADDRESS |
| 00035 | | WRKLEN | EQU | (*-WRKBCTL+3)/4 | LENGTH WORK AREA IN FULLWORDS |

Index

Special Characters

| | | | |
|----------|----|----------|-----|
| #AACTDS | 1 | #EFHDRDS | 92 |
| #ADBDS | 2 | #EFMASDS | 93 |
| #ADEDS | 4 | #EFXDEDS | 94 |
| #ADSDBST | 5 | #EXCDS | 95 |
| #AFGDS | 6 | #EXCFNDS | 96 |
| #AGBDS | 7 | #EXEDS | 97 |
| #AGLDS | 8 | #FDBDS | 98 |
| #AGN1CDS | 9 | #FDEDS | 100 |
| #AGNDS | 10 | #FHEDS | 102 |
| #AGRDS | 11 | #FMDDDS | 104 |
| #AGWADS | 12 | #FMIOWRK | 105 |
| #AIRDS | 18 | #FSRTABL | 106 |
| #AMHDS | 20 | #GABDS | 107 |
| #AMRDS | 21 | #GARDS | 109 |
| #ANCBDS | 22 | #GCMDS | 111 |
| #APGDS | 23 | #GFDDDS | 112 |
| #APLGDS | 25 | #GFEDS | 116 |
| #APREDS | 28 | #GFRDS | 117 |
| #ARDDS | 29 | #GLEDS | 118 |
| #ARSDS | 30 | #GMDDDS | 119 |
| #ASBDS | 31 | #GMEDDS | 120 |
| #ASCDS | 33 | #GPRDS | 121 |
| #ASPDS | 34 | #GRDDDS | 123 |
| #ASQDS | 37 | #GREDS | 126 |
| #ASRDS | 38 | #GRLDS | 129 |
| #ATCDS | 39 | #GSADDS | 130 |
| #AWERRS | 40 | #GSSDDS | 142 |
| #BDSDS | 41 | #GTCDS | 143 |
| #BLEDS | 42 | #GVCDS | 144 |
| #BLPDS | 49 | #GWADS | 145 |
| #BMRBHDS | 50 | #HCHDS | 150 |
| #BPADS | 51 | #HVTDS | 152 |
| #BPXDS | 55 | #INCDS | 153 |
| #BRPLDS | 58 | #LGSDS | 154 |
| #BSBDS | 62 | #LMTDS | 156 |
| #BSPDS | 64 | #LNTDS | 157 |
| #CMEDDS | 66 | #LOBDS | 158 |
| #CMEEQU | 80 | #MAPTDS | 159 |
| #DGBDS | 82 | #MASKDS | 160 |
| #DIBDS | 84 | #MDTDS | 161 |
| #DNBDS | 86 | #MSQDS | 162 |
| #DRNDS | 87 | #MTEDS | 163 |
| #DSDWADS | 88 | #MXDEDS | 164 |
| #DWADS | 89 | #MXSHDS | 165 |
| #DXBDS | 90 | #NAMEDS | 166 |
| | | #OCBDS | 167 |
| | | #OCCLDS | 169 |

```
#OFDDS 170
#OFTDS 171
#OTBXDS 172
#PEXCDS 174
#PFLDS 175
#PICDS 177
#PLFDS 178
#PMEDS 179
#PRBDS 180
#PSADS 181
#PWADS 194
#RACTDS 195
#RBBDS 196
#RDBDS 198
#RDEDS 199
#REXDS 201
#RKEDS 203
#RLODS 204
#RPEDS 205
#RSEDS 206
#RSPDS 207
#RSRTABL 208
#RSTDSD 209
#RVBDS 210
#SACTDS 211
#SDEDS 212
#SELDS 213
#SFEDS 214
#SFTDS 215
#SREDS 216
#SRTDS 219
#SSEDS 220
#STHDS 221
#SWRDS 226
#SYADS 228
#TAEDS 230
#TATUDS 231
#TFEDS 232
#URNDS 233
#VDBDS 234
#WKBCTL 237
```


